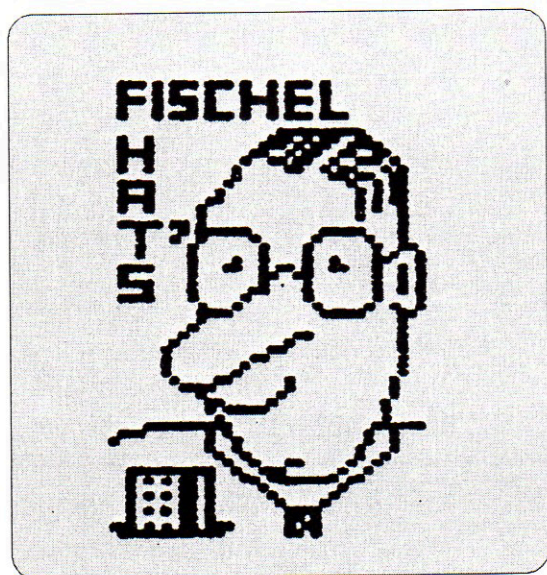


**Datenübertragungs-
Handbuch
für
SHARP-Taschencomputer**



Fischel GmbH

ISBN: 3-924327-63-7

2. überarbeitete Auflage

Bernd Rüter

Do not sale !

=====
C FISCHEL GmbH

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem (Fotokopie, Mikrokopie) oder sonstigem Wege zu vervielfältigen. Es kann keine Haftung für die Richtigkeit der Programme übernommen werden, obwohl sie ausgetestet wurden.

=====



Inhalt

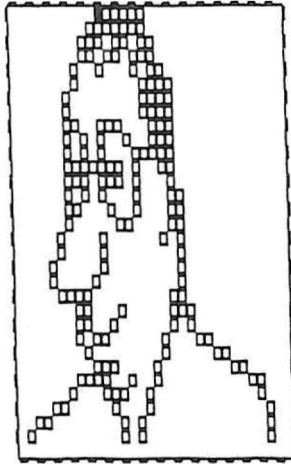
Seite	Titel
2-3	Inhaltsverzeichnis
	SERIELLE SCHNITTSTELLE
4	Vorwort
5	Die serielle Schnittstelle
7	Die Datenformate
8	- Asynchronisches Datenvormat - Synchronisches Datenformat
9	Handshake
	- Hardware - Handshakes - Software - Handshakes
10	
11	Die Form der Übertragung
	- Startbit - Datenbits - Parität
12	- Stopbit
14	- Die Baudrate
	Technik
15	Der Standart
16	- Logikpendel
17	- SN 75188
18	- SN 75189/SN 75189A
19	- Applikationen
20	- MAX 230 - Familie
22	- Applikationen
	Die Normen
23	- Steckerbelegung 25-Pol
25	- Steckerbelegung 9-Pol
27	- Die Bedeutung der Signale
29	Das Verbindungskabel
31	- Das Null-Modem
32	- das richtige Null-Modem
34	- Kabellänge
	PARALLELE SCHNITTSTELLE
35	Vorwort
36	Das Datenformat
38	Elektrische Eigenschaften
	Die Normen
39	- Steckerbelegung 36-Pol
41	- Steckerbelegung IBM 25-Pol
43	- Die Bedeutung der Signale
45	Das Verbindungskabel
46	- Kabellänge
47	Literatur
48	Deutsche ASCII -Tabelle

52	Datenübertragung mit Sharp PC-1500 (A) und Schnittstelle CE 158
54	Daten übertragen mit 2 PC-1500 (A) und CE - 158
55	- Programme übertragen
56	Brother EP44/PC-1500/ CE-158
57	Datenübertragung mit der Schnittstelle RS 232C
	- Vorwort
58	- Allgemeines
	- Die Schnittstelle RS 232C
62	- Checkliste zur Übertragung
63	- Häufig benutzte Zeichensätze
67	Serielle Schnittstelle für den PC-1600
70	Serielle Schnittstelle für den PC-1500/CE-158
73	Serielle Schnittstelle für den PC-1360/CE-130T
76	PC-1500 centronics-Schnittstelle
78	PC-1500 Matext VI/CE158 -Anschluß an centronics Drucker
80	CE-140F Pocket Disk Dire
83	<u>Centronics selbstgemacht</u>
89	<u>RS 232C am Sharp 1401/02</u>
98	Superbestellschein
99	Möglichkeiten der Zusammenarbeit mit der Fischel GmbH
100	Abonnement

Firma
Fischel GmbH
Kaiser-Friedrich-Straße 54a
1000 Berlin 12

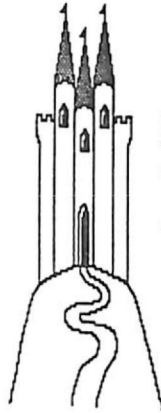


Das große Problem der seriellen Schnittstelle, ist die richtige Verbindung zwischen zwei Geräten, da die Hersteller verschiedene Belegungen verwenden.



Die Datenformate

Die Übertragung der Daten muß bei der seriellen Schnittstelle nach strengen Regeln erfolgen, da sonst die Verständigung nicht klappt. Stellen Sie sich vor, sie Unterhalten sich mit einem Ausländer dessen Sprache Sie nicht verstehen oder Ihr Gesprächspartner spricht so schnell, daß Sie ihm nicht folgen können. Diese Probleme und andere treten auch in der Datenübertragung auf.



Eine wirklich
vornehme
Adresse.....

Datenformate

Asynchrones Datenformat

Dieses Datenformat wird bevorzugt verwendet. Es bedeutet, daß die Bytes (1 Byte = 8 Bit) bitweise (Jedes Bit einzeln) mit bestimmten zeitlichen Längen übertragen werden. Der Empfänger muß also immer auf ein Bit warten und erkennt das erste Bit (Startbit, gehört nicht zu den Daten) nach dem die Datenbits kommen. Sender und Empfänger müssen für die einzelnen Bits die gleichen Zeiten veranschlagen, sonst kommen sie aus dem Tritt. Das Startbit ist also die 'Startklappe' für den Empfänger.

Synchrones Datenformat

Dieses seltene Format fasst mehrere Bytes zu einem Block zusammen. Dem Block wird ein Synchronisationsbyte vorangestellt, sowie ein Endezeichen angefügt. Die Zeichen selbst werden ohne zusätzliche Bits unmittelbar nacheinander übertragen. Dabei müssen die Zeiten für die einzelnen Bits sehr genau eingehalten werden, darum wird zu den Daten auch ein Taktsignal übertragen.

Handshake

Das wichtigste an der Datenübertragung ist das Handshake. Es sorgt dafür, daß keine Zeichen verloren gehen. Wenn der Empfänger nicht schnell genug ist, so kann er den Sender mit dem Handshake stoppen, damit erst die empfangenen Zeichen verarbeitet werden können.

Das Handshake kann mit Hilfe der Steuerleitungen (Hardware) oder mit Steuerzeichen (Software) realisiert werden. Wichtig ist, daß Sender und Empfänger das gleiche Verfahren verwenden.

Hardware-Handshake

Hier werden bestimmte Schnittstellen-Leitungen zur Steuerung der Übertragung verwendet. Hier gibt es verschiedene Leitungen, die später im Buch besprochen werden. Ein Leitungspaar gab dem Verfahren auch den Namen: RTS/CTS-Protokoll. Ein Problem gibt es hier auch, weil manche Leitungen nur von der Software ausgewertet werden und andere Leitungen auch die Hardware steuern. So kann ein Steuersignal mitten in der Übertragung eines Bytes dieses abbrechen und verstümmeln. Dieser Effekt ist von der Schnittstelle/Hardware abhängig und muß einzeln ausgetestet werden.

Datenformate

Software-Handshake

Ein Verfahren wird allgemein als XON/XOFF-Protokoll bezeichnet. Es sieht vor, daß der Empfänger ein XOFF-Zeichen sendet, wenn er die Übertragung anhalten will. Hat er die Zeichen soweit verarbeitet, daß es weitergehen kann, sendet er ein XON, damit der Sender weiterarbeiten kann. Ein kleines Problem ist, daß nach einem XOFF-Zeichen noch ein paar Zeichen vom Sender folgen können. Das liegt an der Hardware und kann nur schwer unterbunden werden.

Ein anderes Verfahren, daß ETX/ACK-Protokoll geht in eine andere Richtung. Hier sendet der Sender eine Anzahl Bytes und schließt sie mit ETX ab. Der Empfänger erkennt am ETX das Ende der Übertragung und kann die Daten verarbeiten. Wenn er fertig ist, sendet er ein ACK zurück und sagt damit, daß er weitere Zeichen empfangen kann. Dabei ist wichtig, daß der Sender nicht mehr Zeichen in einem Block schickt, als der Empfänger im Buffer zwischenspeichern kann.

Darum wurde ein anderes Protokoll entwickelt, daß ACK-Protokoll, bei dem jedes Zeichen vom Sender mit einem ACK von Empfänger beantwortet wird. So kann nach jedem Zeichen abgebrochen werden.

Die Form der Übertragung

Jetzt geht's an Eingemachte: Wie werden die Bytes nun wirklich übertragen, was ist 'seriell', 'asynchron' oder 'Stopbit' ?

Bei der Übertragung wird ein Zeichen (normalerweise ein Byte, es gibt aber auch 5- und 7-Bit-Zeichen) mit Start- und Stopbit versehen, damit der Empfänger sich mit dem Sender abstimmen kann.

Wenn nichts gesendet wird, also auch zwischen zwei Zeichen, wird die Leitung auf '1' (MARK) gesetzt.

Startbit

Das Startbit ist eine '0' (SPACE), damit man es von der freien Leitung ('1') unterscheiden kann. Das Startbit wird vor dem eigentlichen Zeichen gesendet. Der Empfänger wartet auf dies Zeichen und kann die folgenden Bits jetzt abtasten, indem er versucht immer die Mitte der Bits zu treffen, damit die höchste Sicherheit gegeben ist. Die Mitte der Zeichen kann er finden, da er weiß, wie lang (Zeit) ein Bit ist.

Datenbits

Das Zeichen wird vom niederwertigsten zum höchstwertigen Bit einzeln übertragen. Normalerweise ist ein Zeichen 8 Bit lang, es gibt aber auch Formate mit nur 5 oder 7 Bit. Zusätzlich kann bei der 7-Bit-Übertragung das 8.Bit auf '0' oder '1' gesetzt werden. Das ist dann eine 8-Bit-Übertragung.

Parität

Daran schließt sich evtl. ein Paritäts-Bit an, mit dem man Übertragungsfehler im Zeichen

Datenformate

erkennen kann. Es gibt zwei verschiedene Arten der Parität: gerade (even) und ungerade (odd). Das bedeutet, daß die übertragenen Bits (Zeichen + Paritätsbit) zusammen eine gerade oder ungerade Summe von '1' haben müssen, je nach eingestelltem Modus. So ergibt sich für das ASCII-Zeichen '9' das Muster:

'9' ist \$39 ist 00111001

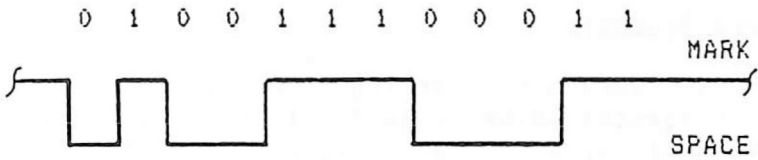
bei ungerader Parität muß das zusätzliche Paritätsbit '1' sein, damit die Summe der '1' ungerade (=5) wird. Die Parity-Prüfung kann auch ausbleiben, dann wird kein Bit angehängt. Beim Empfang wird diese Prüfung auch durchgeführt. Kommt es dabei zu Unstimmigkeiten, ist ein Übertragungsfehler erkannt. Werden bei der Übertragung zwei Bit zerstört, kann der Fehler nicht mehr erkannt werden, weil jetzt die Anzahl wieder stimmt.

Stopbit

Am Schluß des Zeichens wird noch ein oder zwei Stopbit angehängt. Es gibt auch hier wieder einen etwas selteren Modus mit 1,5 Stopbit. Das Stopbit ist eine '1' und wird vom Empfänger erwartet und dient auch als Pause für die Verarbeitung des Zeichens. Man sollte auch daran denken, daß die Wahl von 2 Stopbit die Übertragung verlangsamt aber auch sicherer macht.

Das Format wird oft durch eine sinnreiche Abkürzung angegeben. So bedeutet 8N1 z.B. 8 Datenbit, keiner Paritätsprüfung, und 1 Stopbit. 7E2 ist dann eine Übertragung mit 7 Datenbit, gerader Parität, und 2 Stopbit.

Datenformate

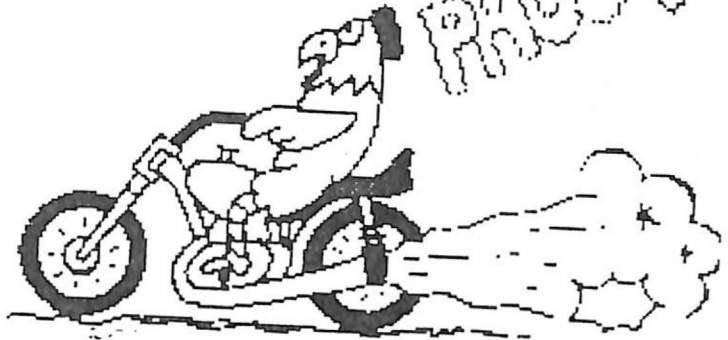


St D0 D1 D2 D3 D4 D5 D6 D7 Pa Stop

Hier sehen Sie ein Beispiel für die Übertragung des Zeichens '9' (ASCII #39) mit 2 Stopbit und even-Parity (gerade)

CE-140F

Row



Datenformate

Die Baudrate

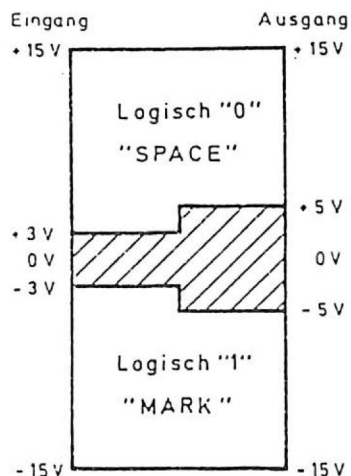
Dies Zauberwort verbirgt nur die Übertragungsgeschwindigkeit in Bit/Sekunde. Im Telefonnetz sind 300 Baud üblich. Es gibt aber auch Ausnahmen mit 1200 Baud. Die maximale Geschwindigkeit ist 19200 Baud. In Sonderfällen geht man noch höher, darunter leidet aber die Sicherheit stark.

Bei dem Übertragungsformat 8N1 werden für jedes Zeichen 10 Bit übertragen: 1 Startbit, 8 Datenbit, 1 Stopbit. Im Telefon werden pro Sekunde 300 Bit übertragen. Das sind dann 30 Zeichen pro Sekunde. Wenn man daran denkt, daß heutige Matrixdrucker locker 200 Zeichen pro Sekunde auf das Papier werfen, ist das wenig, wenn man die Zeichen aber lesen muß ist die Geschwindigkeit doch recht hoch.

Der EIA Standard RS-232-C, die CCITT-Empfehlung V.24/V.28 und die DIN 66020

Im Standard sind folgende elektrischen Eigenschaften festgelegt:

Ausgangs-Spannung 3-7 kOhm Last	+/- 5-15 Volt
Eingangsspannung 3-7 kOhm Last	+/- 3-15 Volt
Ausgangs-Spannung ohne Last	max. +/- 25 Volt
Ausgangs-Spannung für SPACE	> +3 Volt
Ausgangs-Spannung für MARK	< -3 Volt
Leerlaufstrom	5 mA
Eingangswiderstand	3-7 kOhm
Eingangswiderstand VCC=0	min. 300 Ohm
Kurzschlußstrom	max. 500 mA
Flankensteilheit dU/dt	max. 30 V/us
TTL-Ausgang bei offenem Eingang	'1'
TTL-Ausgang bei 300 Ohm gegen Gnd	'1'
TTL-Ausgang bei +3.0V am Eingang	'0'
TTL-Ausgang bei -3.0V am Eingang	'1'
Baud-Rate	max. 20000 Baud



Technik

Logikpegel

Die Eingangsspannung (siehe oben) kann niedriger sein, als die zulässige kleinste Ausgangsspannung, weil auf der Leitung Spannung verloren gehen kann.

Für die beiden Pegel 'MARK' und 'SPACE' gibt es viele Namen. Ich möchte hier ein paar aufführen, damit man Beschreibungen besser verstehen kann.

Logisch 0	Logisch 1
SPACE	MARK
DFF	DN
AUS	EIN
(+)	(-)
POSITIV	NEGATIV
LO	HI
LOW	HIGH
FALSCH	WAHR
START	STOP
ZURÜCKSETZEN	SETZEN
PERFORATION	KEINE PERFORATION

SN75188

Dieser Treiber ist der Standard für serielle Schnittstellen. Praktisch unverwüstlich und billig wird er gerne eingesetzt. Dieser Treiber benötigen aber eine externe Spannungsversorgung von +/- 9-15 Volt.

Die Eingänge des SN75188 sind kompatibel zur TTL- und DTL-Logik. Die Ausgänge sind kurzschlußsicher und entsprechen der RS-232-Vereinbarung.

Hier einige interessante technische Daten:

Versorgungsspannung	+/- 9-15 Volt
Ausgangs-Spannung bei +/- 9 Volt	+/- 7 Volt
Ausgangswiderstand	min. 300 Ohm
Kurzschlußstrom	max. +/- 10 mA
Eingangsspannung für '0'	max. 0,8 Volt
Eingangsspannung für '1'	min. 1,9 Volt

Technik

SN75189, SN75189A

Für diesen Empfänger gilt gleiches wie für den SN75188. Er benötigt aber nur eine Versorgungsspannung von 5 Volt.

Die Ausgänge des SN75189(A) sind kompatibel zur TTL-Logik. Die Eingänge entsprechen der RS-232-Norm.

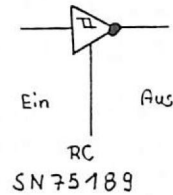
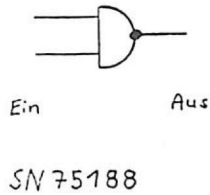
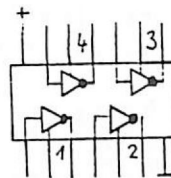
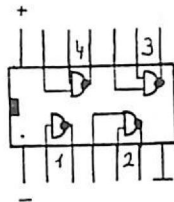
Hier einige interessante technische Daten:

Versorgungsspannung		max. 10 Volt typ 5
Positive Schwellspannung	75189	max. 1,5 Volt
	75189A	max. 2,25 Volt
Negative Schwellspannung		max. 1,25 Volt
Ausgangs-Spannung für '0'		max. 0,45 Volt
Ausgangs-Spannung für '1'		min. 2,6 Volt
Eingangsspannung		max. +/- 30 Volt
Eingangswiderstand		3-7 kOhm

Diese Treiber gibt es auch von anderen Herstellern. Z.B. die Typ MC1488, MC1489 und MC1489A von Motorola.

Applikationen

Der Eingang Response Control dient nur einer Verschiebung der Eingangs-Hysterese oder einer Filterung und wird nicht beschaltet.



Technik

MAX230-Familie

Die MAX230-Familie von MAXIM ist eine kostengünstige Alternative zu herkömmlichen Treibern mit externen Stromversorgungen. Die Treiber erzeugen die nötigen Spannungen von ± 10 Volt mit Hilfe von zwei Kondensatoren durch Chip-interne Spannungsinverter und -verdoppler aus der Versorgungsspannung von 5 Volt. Einige ICs der Familie haben diese Kondensatoren auf dem Chip integriert.

Der verbreitetste Treiber ist hier der MAX232, weil er 2 Eingänge und 2 Ausgänge hat, was für die meisten Anwendungen reicht. Außerdem ist der Preis für dies IC (6/87 ca. 14 DM) durchaus günstig im Vergleich zu einem Netzteil (Netztrafo, Stabilisatoren, Kondensatoren) mit den herkömmlichen Treiber MC1488/89.

Zudem sind die RS232-Eingänge der ICs so konzipiert, daß auch 'normale' TTL-Pegel für einen korrekten Empfang reichen. Die Treiber MC1488/89 benötigen leider mindestens ± 3 Volt.

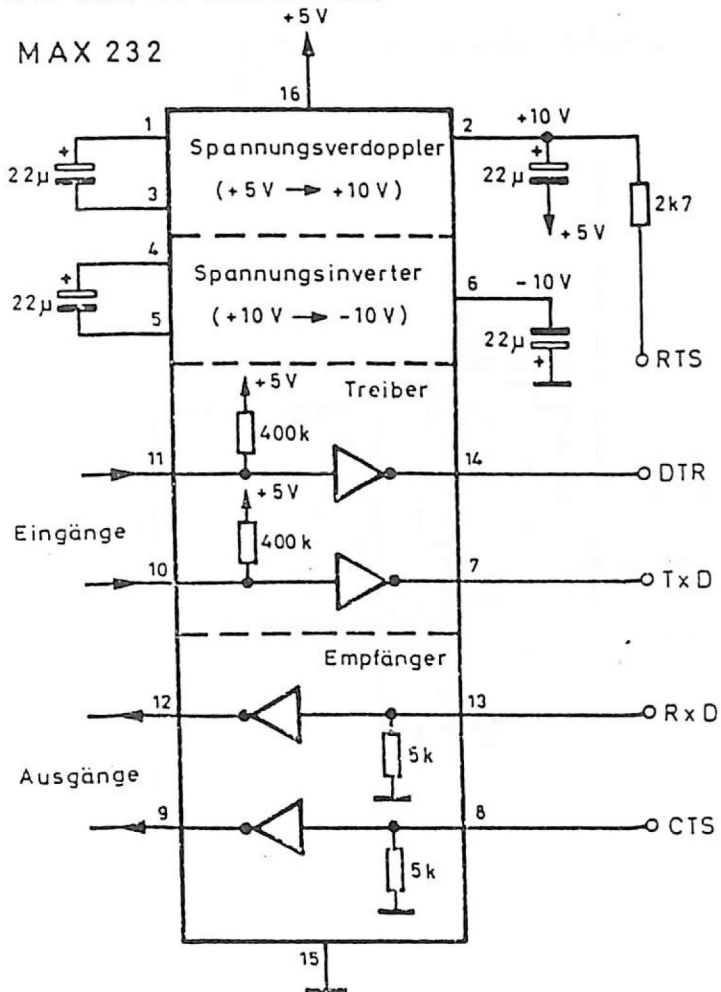
Hier einige interessante technische Daten:

Ausgangs-Spannung (RS-232)	± 9 Volt
Leerlaufstrom	5 mA
RS-232-Eingangsspannung '0'	0,8 - 1,2 Volt
RS-232-Eingangsspannung '1'	1,7 - 2,4 Volt
RS-232-Eingangswiderstand	5 kOhm
RS-232-Kurzschlußstrom	± 10 mA

Die Treiber der MAX230-Familie sind für einen RS-232- und für den V.28-Betrieb geeignet. Sie sind wie üblich in invertierender Logik

aufgebaut. Die TTL-Eingänge und -Ausgänge arbeiten auch mit CMOS-Logik. Die Schaltschwellen und Pull-Ups sind entsprechend ausgelegt. Die RS-232-Eingänge sind gegen Überspannungen bis +/-30 Volt geschützt.

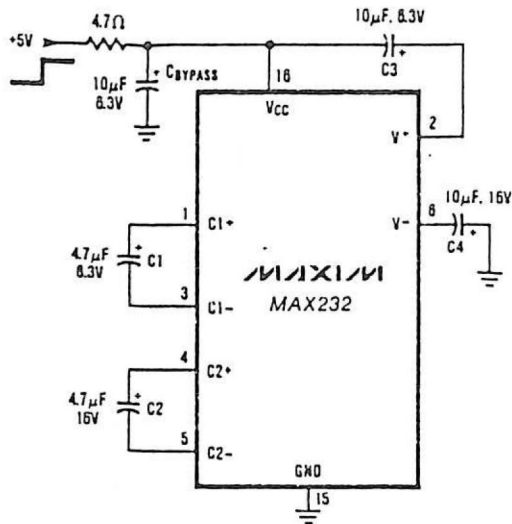
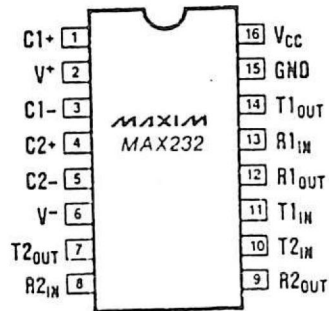
Unbenutzte Eingänge (TTL/CMOS- und RS-232-Seite) sollte man offen lassen um den Stromverbrauch zu minimieren.



Technik

Applikation

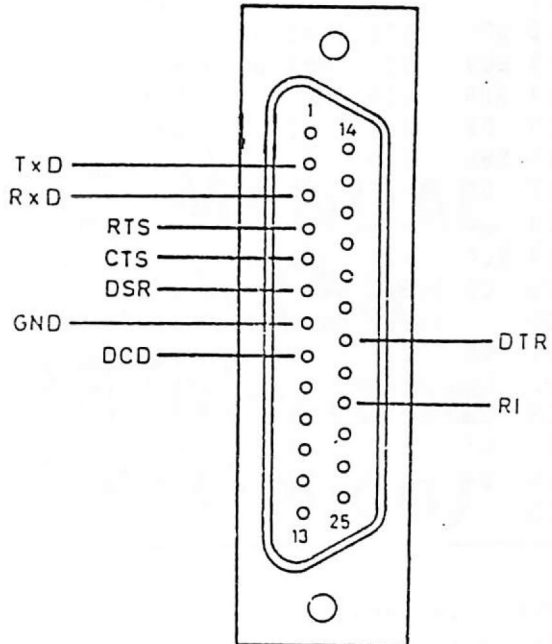
Die Werte der Kondensatoren sind nicht kritisch. Man sollte die Werte aber nicht wesentlich unterschreiten. Den Filter in der Versorgungsleitung sollte man nicht unterschätzen !



Die Normen

In den Normen ist ein 25-poliger SUB-D-Steckverbinder vorgesehen. Es kommen aber auch andere Stecker vor. In jedem Fall soll man das Handbuch konsultieren.

Am DEE soll ein 'männlicher' (Stecker mit Stiften) und am DÜE ein 'weiblicher' (Buchse) Anschluß vorhanden sein.

Belegung des 25-Pol-Steckers nach Normen:

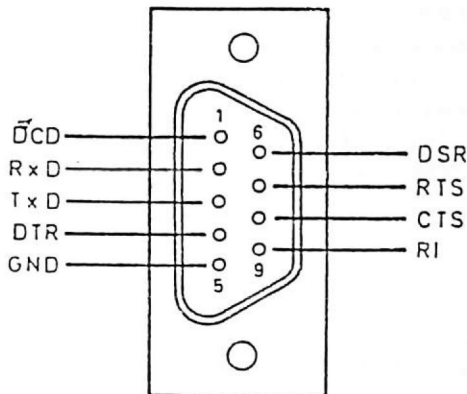
Normen

Pin	RS-232	CCIT V.24	DIN 66020	Beschreibung	DCE	DTE
1	AA	101	E1	Schutzerde		
2	BA	103	D1	TxD Sendedaten	ein	aus
3	BB	104	D2	RxD Empfangsdaten	aus	ein
4	CA	105	S2	RTS Sendeteil einschalten	ein	aus
5	CB	106	M2	CTS Sendebereitschaft	aus	ein
6	CC	107	M1	DSR Betriebsbereitschaft	aus	ein
7	AB	102	E2	Gnd Signalerde		
8	CF	109	M5	DCD Empfangssignalpegel	aus	ein
9				Modemtest		
10				Modemtest		
11				nicht belegt		
12	SCF	122	HMS	SDCD Rückkanal Empf.pegel	aus	ein
13	SCB	121	HM2	SCTS Rückkanal Sendebereit.	aus	ein
14	SBA	118	HD1	STxD Rückkanal Sendedaten	ein	aus
15	DB	114	T2	TxC Sendetakt vom DCE	aus	ein
16	SBB	119	HD2	SRxD Rückkanal Empfangsdat.	aus	ein
17	DD	115	T4	RxC Empfangsschrittakt	aus	ein
18				nicht belegt		
19	SCA	120	HS2	SRTS Rückkanal Sendteil an	ein	aus
20	CD	108.2	S1.2	DTR Terminal betriebsber.	ein	aus
20		108.1	S1.1	Übertragungsleitg. an	ein	aus
21	CG	110	M6	Empfangsgüte	aus	ein
22	CE	125	M3	RI Ankommender Ruf	aus	ein
23	CH	111	S4	Übertrg-gschw.vom DTE	ein	aus
23	CI	112	M4	Übertrg-gschw.vom DCE	aus	ein
24	DA	113	T1	Sendetakt vom DTE	ein	aus
25				nicht belegt		

Die Leitungen 9 und 10 werden oft bei Akkustikkopplern als Stromversorgungen verwendet.

Belegung des IBM-9-Pol-Steckers:

Neben dem obigen 25-poligen-Stecker ist heute auch die Belegung des Industrie-Standarts wichtig.



SERVICE MANUAL

MODEL CE-158

RS-232C Interface

(PC-1500 Option)

1. Introduction
2. Specification
3. System configuration ...
4. Block diagram
5. Circuit discription
6. Lsi discription
7. IC pin connection
8. Circuit diagram
9. Color chart
10. Parts Guide/Parts list

Normen

Pin	Beschreibung	DCE	DTE
1	DCD Empfangssignalpegel	aus	ein
2	RxD Empfangsdaten	aus	ein
3	TxD Sendedaten	ein	aus
4	DTR Terminal betriebsbereit	ein	aus
5	Gnd Signalerde		
6	DSR Betriebsbereitschaft	aus	ein
7	RTS Sendeteil einschalten	ein	aus
8	CTS Sendebereitschaft	aus	ein
9	RI Ankommender Ruf	aus	ein



Die Bedeutung der Steuersignale:

Auf den Steuerleitungen bedeutet ein SPACE eine Bereitschaftsmeldung.

DTR

Betriebsbereitschaft des DTE als Aufforderung, die Übertragungsleitung einzuschalten.

DSR

Betriebsbereitschaft des DCE als Antwort, daß das Gerät eingeschaltet ist und die Übertragungsleitung geschaltet ist.

RTS

Aufforderung des DTE den Sender zu aktivieren, damit der DTE Daten senden kann.

CTS

Antwort des DCE auf die Aufforderung damit Daten übertragen werden können.

DCD

Signal, daß die Gegenstation mit ausreichendem Pegel sendet, um falsche Daten zu erkennen.

RI

Ankommender Ruf, der DCE gibt beim 'Klingeln' des Telefons dies Signal.

TxC

Sendeschrifttakt vom DCE für Synchrone Übertragung.

RxC

Empfangsschrifttakt vom DCE für Synchrone Übertragung.

Normen

Gnd

Signalmasse für alle Leitungen.

Schutzerde

Dies Signal sollte nicht mit Gnd verbunden werden. Es ist mit dem Gehäuse zur Abschirmung verbunden. Eine Abschirmung im Kabel sollte man an einem Ende mit diesem Anschluß verbinden, sonst kommt es zu Erdschleifen, die sehr störend sein können.



Das Verbindungskabel

Das Verbindungskabel für eine serielle Verbindung ist immer ein Problem. Durch die vielfältigen Realisationen der seriellen Schnittstelle gibt es immer wieder Schwierigkeiten.

Zuerst muß man aber feststellen, von welchem Typ beide Geräte sind. Dazu nimmt man zuerst das Handbuch zu einem Gerät und sieht nach, was dort angegeben ist. Wurde Pin 2 mit TxD als Ausgang gekennzeichnet, ist es ein DEE (DTE). Wenn Pin 2 mit TxD als Eingang gekennzeichnet ist, handelt es sich um ein DÜE (DCE). Entscheidend ist hier die Richtung.

Falls das Handbuch sich über diese Angaben ausschweigt, kann man mit einem einfachen Meßgerät die Belegung feststellen. An einem Eingang muß man ca. 0 Volt messen, bei einem Ausgang aber liegt eine deutliche Spannung an.

Verbindungskabel

Es gibt also zwei Sorten von Verbindungen. Auf der einen Seite sind alle 'richtigen' zwischen einem DEE (DTE) und dem DÜE (DCE), auf der anderen Seite kommt dann der Rest, die Verbindung zwischen zwei gleichen Geräten. Die Norm sieht aber nur eine Verbindung der ersten Sorte vor. Bei den anderen Verbindungen muß man etwas schummeln.

Soweit wir eine Verbindung des ersten Typs haben, werden nur alle Pins eins-zu-eins miteinander verbunden. D.h., daß Pin 2 vom DEE (DTE) mit Pin 2 vom DÜE (DCE) usw. verbunden wird.

Das ergibt folgendes Kabel:

TxD	2	—————	2	TxD
RxD	3	—————	3	RxD
RTS	4	—————	4	RTS
CTS	5	—————	5	CTS
DSR	6	—————	6	DSR
GND	7	—————	7	GND
DCD	8	—————	8	DCD
DTR	20	—————	20	DTR
RI	22	—————	22	RI

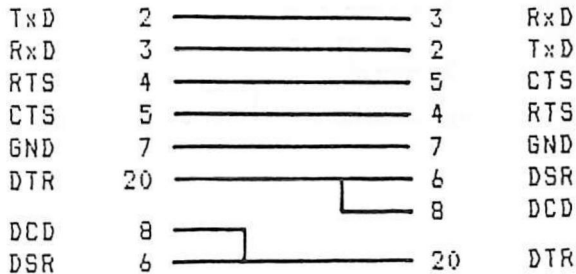
Das Kabel kann auch einfacher ausfallen, wenn kein Hardware-Handshake benötigt wird, und der normale Software-Handshake von beiden Geräten unterstützt wird.

Das Null-Modem

Für den zweiten Typ wird es schon schwieriger. Zwar sind hier die Richtungen für die einzelnen Signale (meißt) eindeutig, weil es sich um DEEs (DTE) handelt, jedoch muß hier auch immer das Handbuch zu Rate gezogen werden.

Als gängige Verbindung hat sich hier das 'Null-Modem' oder auch das 'gekreuzte' Kabel durchgesetzt. Diese Verbindung ist zwar grundlegend falsch, funktioniert aber oft, weil die Hersteller langsam darauf eingegangen sind.

Das Null-Modem:

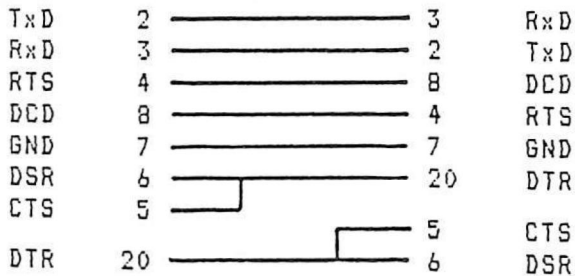


Verbindungskabel

Das richtige Null-Modem

Der Fehler liegt in der Verbindung des Paares RTS/CTS. Die Leitung CTS dient dazu, den Sender zu sperren, wenn der Empfänger nicht bereit ist. In diesem Kabel wird das Signal CTS mit RTS verbunden. Das bedeutet, daß die Sendeaufforderung des einen Geräts als Steuersignal für das andere geschaltet ist. Normalerweise müßte CTS mit DCD verbunden sein, damit die Funktion einen Sinn ergibt.

Das richtige Null-Modem:



Verbindungskabel

Funktioniert die Verbindung jetzt noch nicht, kann es daran liegen, daß ein Gerät ein Signal nicht ausgibt, das andere dies aber dringend braucht. Z.B. kann am DÜE (DCE) das Signal CTS fehlen, das die DEE (DTE) dringend für die Übertragung benötigt.

In diesen Fällen muß man die fehlenden Signale am Empfänger durch andere Signale ersetzen. Man kann z.B. an den Eingang direkt eine positive Spannung anlegen, die am Stecker evtl. zur Verfügung steht oder man nimmt einen Ausgang des Gerätes, der immer auf 'SPACE' steht. Dazu sind im Normalfall immer die Gegenstücke der betreffenden Eingänge brauchbar, so kann man ein fehlendes CTS am Gerät mit dem RTS verbinden. Das Signal CTS sollte dann nicht mehr zum anderen Gerät geführt werden.

Verbindungskabel

Kabellänge

Das Kabel für eine Verbindung darf nicht zu lang sein, da sonst die Übertragung stark gestört sein kann. Ein Richtwert aus der Norm für Koaxial-Kabel ist folgende Formel:

$$\text{Kabellänge in m} = \frac{1\ 000\ 000\ \text{m}}{\text{Baudrate}}$$

maximal jedoch nur 1000m.

Man sieht sofort, daß eine Übertragung mit 9600 Baud nur eine Länge von 100m zuläßt.

In der Wirklichkeit wird aber mit normalen Kabel gearbeitet, das maximal normal abgeschirmt oder evtl. verdrillt ist. Darum muß man die Länge noch wesentlich verkürzen. Kabellängen von etwa 20m zwischen zwei Geräten sind wohl eine gute Richtlinie. Die Norm läßt eine Länge bis 30,5m zu. Bei hohen Geschwindigkeiten (>9600 Baud) wird diese Länge schon kritisch.

Die Parallele Schnittstelle

Die parallele Schnittstelle ist ein Standard für den Anschluß von Druckern an einen Computer.

Die Schnittstelle kann also nicht in beide Richtungen verwendet werden.

Die Firma CENTRONICS hat den Standard für ihre Drucker geschaffen, als andere Hersteller noch mit den Anschlüssen kämpften. Der Standard hat sich relativ schnell durchgesetzt und ist im Gegensatz zur seriellen Schnittstellen-Norm ohne größere Probleme.

Nur eine einzige Änderung hat die Firma EPSON eingeführt.

Datenformate

Das Datenformat

Die parallele Datenübertragung verlangt keine so aufwendigen Formate wie die serielle Übertragung. Hier genügt eine Festlegung der Reihenfolge der Handshake-Signale.

Aber erst mal zu einfachen Dingen. Die Schnittstelle hat acht Datenleitungen. Da der normale ASCII-Zeichensatz nur 7 Datenleitungen benötigt, wird das 8.Bit von den Druckern unterschiedlich interpretiert. Dazu kommen jetzt noch 3 Leitungen, die die Übertragung steuern. Die Bedeutung der Leitungen wird in einem anderen Kapitel erklärt. Hier möchte ich die zeitliche Reihenfolge der Signale darstellen (siehe auch nächste Seiten).

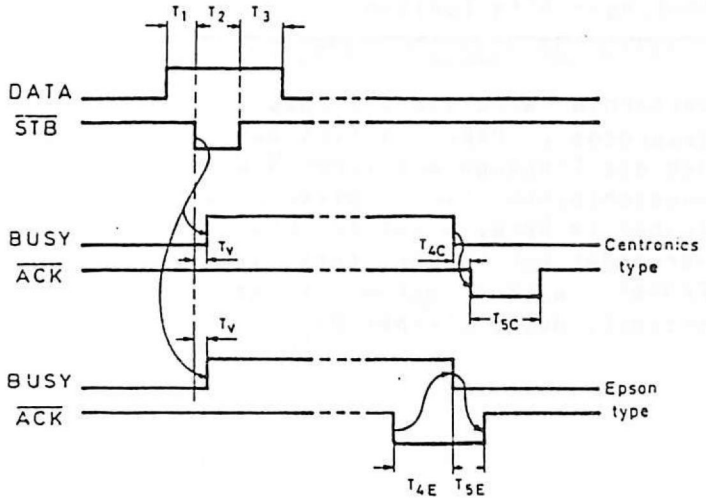
Die Übertragung wird durch das Anlegen des Datenbyte an die Datenleitungen gestartet. Die Übernahme geschieht mit dem Strobesignal, das der Computer auch gibt. Der Drucker setzt danach sein Busy-Signal und verarbeitet das Byte. Am Ende wird, abhängig von der verwendeten Norm, das Busy-Signal zurückgenommen und mit Acknowledge die Übernahme quittiert.

T_1, T_2 und $T_3 = \text{min. } 500\text{ns}$

$T_V = \text{max. } 500\text{ns}$

$T_{4C} = 0 \dots 10\mu\text{s}$ $T_{4E} = 7\mu\text{s}$

$T_{5C} = 5 \dots 30\mu\text{s}$ $T_{5E} = 5\mu\text{s}$



Technik

Elektrische Eigenschaften

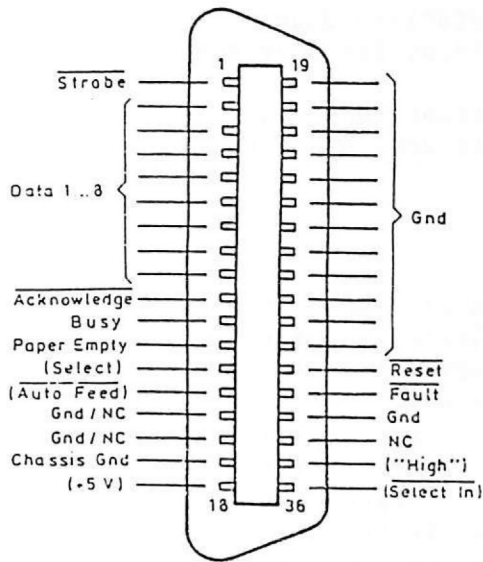
Im Standard sind folgende elektrischen Eigenschaften festgelegt:

Ausgangs-Spannung für '0'	max. 0,8 Volt
Ausgangs-Spannung für '1'	min. 2,4 Volt
Anstiegs- Abfallzeiten	max. 0,2µ sek

Weiterhin wird von TTL-kompatiblen Eingängen gesprochen. Dabei sollte man daran denken, daß die Eingänge mit einem Widerstand an +5V angeschlossen sind. Diese Widerstände waren früher im Bereich von 470 Ohm bemessen, heute verwendet man in der Regel 3,3k bis 4,7k Ohm. Früher wurden darum Ströme bis zu 7mA erzielt. Heute bleiben die Ströme bei 2mA.

Die Normen

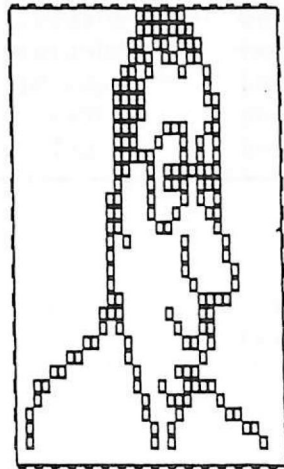
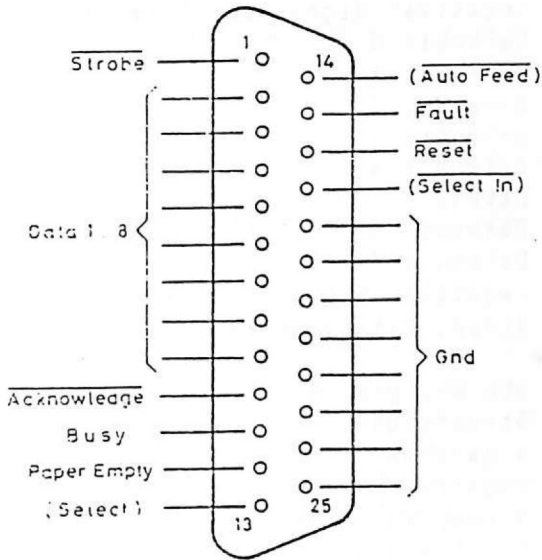
Ursprünglich war ein 36-poliger Steckverbinder vorgesehen. Es kommen aber auch andere Stecker vor. So wird beim IBM-PC ein 25-poliger-SUB-D-Steckverbinder verwendet. In jedem Fall soll man das Handbuch konsultieren.

Belegung des 36-Pol-Steckers nach Standard:

Normen

Pin	Beschreibung	Computer	
1	Strobe	negatives Signal zur Datenübernahme	aus
2	Data 0	Datenbit 0	aus
3	Data 1	Datenbit 1	aus
4	Data 2	Datenbit 2	aus
5	Data 3	Datenbit 3	aus
6	Data 4	Datenbit 4	aus
7	Data 5	Datenbit 5	aus
8	Data 6	Datenbit 6	aus
9	Data 7	Datenbit 7	aus
10	Acknowl.	negatives Signal zur Bestätigung	ein
11	Busy	Stoppt Datenübertragung	ein
12	Papierende		ein
13	Select	Steuersignal für Drucker	aus
14	Auto Feed	Steuersignal für Drucker	aus
15	Gnd/NC		
16	Gnd/NC		
17	Schutzerde		
18	+5V/NC		
19	Gnd	Rückleitung für Strobe	
20	Gnd	Rückleitung für Datenbit 0	
21	Gnd	Rückleitung für Datenbit 1	
22	Gnd	Rückleitung für Datenbit 2	
23	Gnd	Rückleitung für Datenbit 3	
24	Gnd	Rückleitung für Datenbit 4	
25	Gnd	Rückleitung für Datenbit 5	
26	Gnd	Rückleitung für Datenbit 6	
27	Gnd	Rückleitung für Datenbit 7	
28	Gnd	Rückleitung für Acknowledge	
29	Gnd	Rückleitung für Busy	
30	Gnd	Rückleitung für Papierende	
31	Reset	negatives Signal zum Initialisieren	aus
32	Fault	negatives Signal bei Fehler	ein
33	Gnd		
34	NC		
35	High	High-Signal für alle Zwecke	
36	Select in	Steuersignal für Drucker	

Belegung des IBM-25-Pol-Steckers:



Normen

Pin	Beschreibung	Computer
1	Strobe	negatives Signal zur Datenübernahme aus
2	Data 0	Datenbit 0 aus
3	Data 1	Datenbit 1 aus
4	Data 2	Datenbit 2 aus
5	Data 3	Datenbit 3 aus
6	Data 4	Datenbit 4 aus
7	Data 5	Datenbit 5 aus
8	Data 6	Datenbit 6 aus
9	Data 7	Datenbit 7 aus
10	Acknowl.	negatives Signal zur Bestätigung ein
11	Busy	Stoppt Datenübertragung ein
12	Papierende	ein
13	Select	Steuersignal für Drucker aus
14	Auto Feed	Steuersignal für Drucker aus
15	Fault	negatives Signal bei Fehler ein
16	Reset	negatives Signal zum Initialisieren aus
17	Select in	Steuersignal für Drucker
18	Gnd	Rückleitung
19	Gnd	Rückleitung
20	Gnd	Rückleitung
21	Gnd	Rückleitung
22	Gnd	Rückleitung
23	Gnd	Rückleitung
24	Gnd	Rückleitung
25	Gnd	Rückleitung

Die Bedeutung der Steuersignale:

Datenbits

Auf diesen Leitungen liegen die zu übertragenden Bits an.

Strobe

Dies Signal (normal high) wird nach Anliegen (min. 500ns) der Daten kurz (min. 500ns) low, damit die Daten vom Empfänger angenommen werden.

Acknowledge

Dies Signal (normal high) wird kurz (5..30us) low wenn das Byte verarbeitet wurde.

Centronics-Norm: Das Signal beginnt kurz (0..10us) nach der fallenden Busy-Flanke.

EPSON-Norm: Das Signal beginnt kurz (ca. 7us) vor der fallenden Busy-Flanke.

Busy

Dies Signal (normal high) wird kurz (max. 500ns) nach fallender Strobe-Flanke aktiv und zeigt an, daß das Byte verarbeitet wird; der Drucker keine Zeichen empfangen kann.

Papierende (Paper Empty)

Dies Signal (normal low) wird high, wenn kein Papier im Drucker ist.

Auto Feed

Dies Signal dient der Steuerung des automatischen Zeilenfordschubs (LF) bei Empfangen eines Wagenrücklaufs (CR). Ein low auf dieser Leitung schaltet das automatische LF ein.

Reset

Wenn dies Signal (normal high) für min. 50us auf low gelegt wird, wird der Drucker neu normiert (wie nach jedem Einschalten).

Normen

Fault

Wenn ein Fehler (Papierende, OFF-Line, ...) auftritt, geht dies Signal (normal high) auf low.

High

Diese Leitung führt ein high-Pegel (über einen Widerstand).

Select In

Ein high auf dieser Leitung (normal low) ermöglicht die Selektion des Druckers über die ASCII-Zeichen DC1/DC3. Das dient zur Ansteuerung mehrerer Drucker über ein Kabel.

Gnd

Signalmasse für alle Leitungen.

Schutzerde

Dies Signal sollte nicht mit Gnd verbunden werden. Es ist mit dem Gehäuse zur Abschirmung verbunden. Eine Abschirmung im Kabel sollte man an einem Ende mit diesem Anschluß verbinden, sonst kommt es zu Erdschleifen, die sehr störend sein können.

Das Verbindungskabel

Ein Verbindungskabel für eine parallele Schnittstelle ist sehr einfach. Man braucht nur alle Signale entsprechend verbinden. Dabei gibt es nur Probleme mit den Belegungen der Stecker.

Als Material ist paarweise verdrehtes Kabel am besten. Heutzutage reicht wegen der geringeren Störungen auch schon Flachbandkabel, wenn jede zweite Ader auf Massepotential liegt.

Masse ist sowieso noch ein Problem. 'Gnd' als elektrische Signalmasse darf man nicht mit 'Chassis Gnd' verbinden. Außerdem ist Gnd an den Geräten sehr verschieden angeschlossen. Da hilft nur ein genaues Studium der Handbücher.

Bei verdrehten Adern soll man eine Ader an ein Signal und die andere Ader jeweils an Masse anschließen. Bei Flachbandkabel und Quetschverbindern liegt jede zweite Leitung zwischen den Datenleitungen von selbst auf Masse.

'+5V' wird bei manchen Herstellern als Versorgungsspannung für Interface-Schaltungen zur Verfügung gestellt. Auf keinen Fall darf man diese Leitung zwischen zwei Geräten verbinden. Gleiches gilt für die Leitung 'High'.

Verbindungskabel

Kabellänge

Die Kabellänge wird immer mit 'möglichst kurz' beschrieben. Längen von max. 2 bis 3m sollte man nicht überschreiten.



Do not sale !

Literatur:

c't Magazin für Computertechnik Ausgabe 12/86
vom Verlag Heinz Heise GmbH in D-3000 Han-
nover 51

MAXIM +5V Powered RS-232 Drivers/Receivers
Datenblatt der Firma Spezial Elektronik in
3062 Bückeburg

Pocket Guide Band 2 Lineare Integrierte
Schaltungen Ausgabe 1985 von Texas
Instruments Deutschland GmbH in 8050 Freising
Integrated Linear Circuits Ausgabe 1985 von
Texas Instruments Deutschland GmbH in 8050
Freising

Schnittstellen-Handbuch von J.Elsing und
A.Wiencek vom iwt Verlag

Hacker-Handbuch für SHARP-Computer von Sven
Nimsgarn und Bernd Rüter der Fischel GmbH in
Berlin

Tabelle

Deutsche ASCII-Tabelle (nach DIN 66003)

Dez	Hex	Binär	ASCII
0	00	00000000	NUL ^@ Null (Füller)
1	01	00000001	SOH ^A Anfang des Kopfes
2	02	00000010	STX ^B Anfang des Textes
3	03	00000011	ETX ^C Ende des Textes
4	04	00000100	EDT ^D Ende der Übermittlung
5	05	00000101	ENQ ^E Stationsaufforderung
6	06	00000110	ACK ^F Positive Rückmeldung
7	07	00000111	BEL ^G Glocke
8	08	00010000	BS ^H Rückschritt /links
9	09	00010001	HT ^I Horizontaler Tabulator
10	0A	00010010	LF ^J Zeilenvorschub /runter
11	0B	00010011	VT ^K Vertikaler Tabulator /hoch
12	0C	00011000	FF ^L Seitenvorschub
13	0D	00011001	CR ^M Wagenrücklauf
14	0E	00011010	SO ^N Dauerumschaltung (Shift-Out)
15	0F	00011011	SI ^O Rückschaltung (Shift-In)
16	10	00100000	DLE ^P Datenübertragungsumschaltung
17	11	00100001	DC1 ^Q Gerätesteuerszeichen 1
18	12	00100010	DC2 ^R -"- 2
19	13	00100011	DC3 ^S -"- 3
20	14	00100100	DC4 ^T -"- 4
21	15	00100101	NAK ^U Negative Rückmeldung
22	16	00100110	SYN ^V Synchronisierung
23	17	00100111	ETB ^W Ende des Datenübertragungsbl.
24	18	00110000	CAN ^X Ungültig (Cancel)
25	19	00110001	EM ^Y End.d.Aufzeich. (End of Medium)
26	1A	00110010	SUB ^Z Substitutionszeichen
27	1B	00110011	ESC ^[Code-Umschaltung (Escape)
28	1C	00111000	FS ^\ Hauptgrup.-Trennz. (File-Sep.)
29	1D	00111001	GS ^] Gruppen-Trennz. (Group-Sep.)
30	1E	00111010	RS ^^ Untergr.-Trennz. (Record-Sep.)
31	1F	00111011	US ^_ Teilgr.-Trennz. (Unit-Sep.)

Tabelle

32	20	0100000	Space	
33	21	0100001	!	
34	22	0100010	"	
35	23	0100011	#	Nummernzeichen
36	24	0100100	\$	Nationales Währungszeichen
37	25	0100101	%	
38	26	0100110	&	Kommerzielles Und
39	27	0100111	'	Apostroph
40	28	0101000	(
41	29	0101001)	
42	2A	0101010	*	
43	2B	0101011	+	
44	2C	0101100	,	
45	2D	0101101	-	
46	2E	0101110	.	
47	2F	0101111	/	
48	30	0110000	0	
49	31	0110001	1	
50	32	0110010	2	
51	33	0110011	3	
52	34	0110100	4	
53	35	0110101	5	
54	36	0110110	6	
55	37	0110111	7	
56	38	0111000	8	
57	39	0111001	9	
58	3A	0111010	:	
59	3B	0111011	;	
60	3C	0111100	<	
61	3D	0111101	=	
62	3E	0111110	>	
63	3F	0111111	?	

Tabelle

64	40	1000000	S
65	41	1000001	A
66	42	1000010	B
67	43	1000011	C
68	44	1000100	D
69	45	1000101	E
70	46	1000110	F
71	47	1000111	G
72	48	1001000	H
73	49	1001001	I
74	4A	1001010	J
75	4B	1001011	K
76	4C	1001100	L
77	4D	1001101	M
78	4E	1001110	N
79	4F	1001111	D
80	50	1010000	P
81	51	1010001	Q
82	52	1010010	R
83	53	1010011	S
84	54	1010100	T
85	55	1010101	U
86	56	1010110	V
87	57	1010111	W
88	58	1011000	X
89	59	1011001	Y
90	5A	1011010	Z
91	5B	1011011	Ä
92	5C	1011100	ö
93	5D	1011101	ü
94	5E	1011110	^
95	5F	1011111	-

International: @ Kommerz. á

International: [
 International: \
 International:]
 Zirkumflex
 Unterstreichung

Tabelle

96	60	1100000	`	Gravis
97	61	1100001	a	
98	62	1100010	b	
99	63	1100011	c	
100	64	1100100	d	
101	65	1100101	e	
102	66	1100110	f	
103	67	1100111	g	
104	68	1101000	h	
105	69	1101001	i	
106	6A	1101010	j	
107	6B	1101011	k	
108	6C	1101100	l	
109	6D	1101101	m	
110	6E	1101110	n	
111	6F	1101111	o	
112	70	1110000	p	
113	71	1110001	q	
114	72	1110010	r	
115	73	1110011	s	
116	74	1110100	t	
117	75	1110101	u	
118	76	1110110	v	
119	77	1110111	w	
120	78	1111000	x	
121	79	1111001	y	
122	7A	1111010	z	
123	7B	1111011	ä	International: {
124	7C	1111100	ö	International:
125	7D	1111101	ü	International: }
126	7E	1111110	ß	International: ~ Überstreich.
127	7F	1111111	DEL	Löschen

Datenübertragung mit SHARP PC 1500(A)
und Schnittstelle CE 158 von SHARP
von Sven Nimsgarn

Ich möchte mit diesem Bericht diejenigen SHARP PC 1500(A) Besitzer ansprechen, die eine Schnittstelle CE 158 Ihr Eigen nennen. Ich werde einige Möglichkeiten der Datenübertragung im seriellen Betrieb dalegen. Ich hoffe Sie bekommen einige Anregungen und versuchen selbst etwas zu Experimentieren mit der CE 158. Ich wünsche viel Erfolg.

Allgemeines zur Datenübertragung

Seriellles Signal

Stop-Bit	Start-Bit	Daten-Bits						Parität Bit	Stop Bit
		0	1	2	3	4	(5,6,7)		

← Übertragungsrichtung →

So sieht ein übertragenes Byte z.B. aus. Seriell bedeutet, daß Byte für Byte hintereinander übertragen werden. 1 Byte = 8Bit

Baud-Rate: (dt.Übertragungsgeschwindigkeit) Die Baud-Rate steht für die Anzahl der pro Sekunde übertragenen Bits, einschließlich Start-,Daten-,Paritäts-, und Stop-Bit.

Zeichenlänge: Anzahl der Bits zum Übertragen eines Zeichens. ASCII zum Beispiel braucht 8 Bits für ein Zeichen.

Paritäts-Bit: Ein Bit wird an das letzte Daten-Bit angehängt. Zur Überprüfung der Daten kann gleicher/ungleicher oder nichtparitätischer Betrieb gewählt werden.

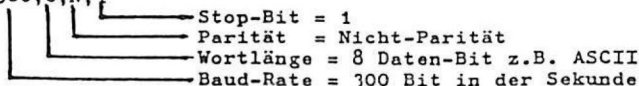
Stop-Bit(s): Ans Daten-Bit bzw. Paritäts-Bit angehängte (s) Bit. Die üblichen Längen sind 1 oder 2 Stop-Bits.

Programmieren der CE 158 Schnittstelle

SETCOM Anweisung kann manuell oder im Programm verwendet werden.

folgende Parameter sind beim Einschalten der CE 158 vorgegeben

SETCOM 300,8,N,1



Eingabe-Möglichkeiten bei SETCOM

Baud-Rate: 50,100,110,200,300,600,1200,2400 je höher die Baud-Rate desto schneller geht die Übertragung von statten

Wortlänge: 5,6,7,8 Daten-Bits für ASCII braucht man 8 Daten-Bits um ein Zeichen zu Übertragen.

Parität: N=Nicht, E=Gleich, O=Ungleich

Stop-Bit(s): 1 oder 2 je nach Schnittstelle verschieden. Die Stop-Bit(s) dienen dazu damit die Schnittstelle weiß wann ein Zeichen zuende ist.

Wollen Sie nun wissen was in der Anweisung steht, so können Sie dieses mit der Anweisung COM\$ abfragen.

Beispiel Rechner und Schnittstelle neu eingeschaltet. COM\$ "ENTER" auf dem Display erscheint 300,8,N,1. geben Sie nun SETCOM 1200,7,N,2 "ENTER" ein und wollen nun die Werte abfragen mit COM\$ "ENTER" so erscheint 1200,7,N,2 versuchen Sie es einmal.

SETDEV: dieses Kommando kann manuell oder Programmgesteuert eingesetzt werden. SETDEV legt die INPUT/OUTPUT Anweisung der CE 158 fest. Das heißt, es wird festgelegt ob eine Eingabe oder Ausgabe auf der Schnittstelle erfolgen soll.

Zuweisung	Eingabe/Ausgabe	Rechnerbefehl	Anwendung
KI	Eingabe	INPUT	eine/mehrere Variablen können über die Schnittstelle in den Rechner eingelesen werden
DO	Ausgabe	PRINT	es werden Variablen über die Schnittstelle ausgegeben
PO	Ausgabe	LPRINT, LLIST	es können Zeichenfolgen oder ganze Programme aus der Schnittstelle ausgegeben werden z.B. auf einen externen Drucker oder Großrechner wo diese dann auf dem Monitor erscheinen
CI	Eingabe	CLOAD, INPUT ≠, MERGE, CLOAD M	es können ganze Prg. von einem SHARP PC 1500(A) zu einem ANDEREM übertragen werden. (siehe Beispielprogramm) was nicht im CE 158 Handbuch steht, ist das man auch Maschinenprogramme übertragen kann
CO	Ausgabe	CSAVE, PRINT ≠, CSAVE M	es werden Prg. über die Schnittstelle ausgegeben

WICHTIG !!!

Die SETDEV-Anweisung muß immer programmiert werden. Das bedeutet, daß die Anweisung beim Einschalten des Rechners und Schnittstelle immer gleich "Ø" b.z.w. leer ist. Also bevor Sie etwas übertragen wollen immer SETDEV KI,DO,PO,CI oder CO entsprechend zuweisen.

DEV\$: mit DEV\$ können Sie die Funktion SETDEV abfragen. (siehe auch COM\$)

OUTSTAT: dient zur Verständigung zwischen zwei Rechnern. Ich habe herausgefunden, daß immer nur der Empfänger OUTSTAT Ø haben muß wenn man Programme oder Variablen zwischen zwei SHARP PC 1500(A) hin- und her transferieren will. (siehe auch Beispiel-Prg.)

ACHTUNG !!! BEACHTEN !!! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Wollen Sie Daten b.z.w. Programme durch die Schnittstelle übertragen, so muß darauf geachtet werden, daß immer die Empfangsstation zuerst gestartet werden sollte.

Denn auf Grund von Installationen braucht der Sender eine gewisse Zeit um betriebsbereit zu sein. Also erst Empfänger-Prg. starten und dann senden.

Daten übertragen mit zwei
SHARP PC 1500(A) und CE 158

Ich habe zwei SHARP PC 1500(A) mit CE 158 Schnittstelle einfach mit einem Kabel über die seriellen Schnittstellen miteinander verbunden. Es ist nur darauf zu achten, daß die Sende- und Empfangsleitungen gekreuzt werden.

Variablen übertragen von SHARP PC 1500(A) zu SHARP PC 1500(A)

<u>Senderprogramm-Listing</u>	<u>Erklärung-Senderlisting</u>
10:"A"REM **SENDE R PC-1500->>PC 1500**	Start mit DEF A
20:REM 30:SETCOM 300,8,N ,1	SETCOM 300 Baud, 8 Daten-Bits, Nicht-Parität, 1 Stop-Bit
40:REM 50:SETDEV 00 60:REM	es wird eine Variable gesendet mit PRINT darum 00 (siehe Tabelle)
70:BEEP 1:INPUT " Variable:";A\$	Eingabe der Variablen in A\$
80:REM 90:PRINT A\$ 100:REM 110:END	an dieser Stelle wird nun durch die Anweisung PRINT die Variable A\$ durch die CE 158 Schnittstelle an den zweiten Rechner gesendet

<u>Empfänger-Listing</u>	<u>Erklärung-Empfängerlisting</u>
10:"B"REM **EMPPFA ENGER PC 1500- >>PC 1500**	Start mit DEF B
20:REM 30:OUTSTAT 0 40:REM	die Schnittstelle wird zum Daten- empfang vorbereitet (engl. Handshake)
50:SETCOM 300,8,N ,1 60:REM	Übertragungsparameter wie Sender, sonst kein ordnungsgemäßer Empfang
70:SETDEV KI 80:REM	es soll eine Variable empfangen werden mit INPUT darum KI (siehe Tabelle)
90:INPUT "Variabl e:";A\$	an dieser Stelle wird nun die empfang- ene Variable in A\$ geladen
100:REM 110:SETDEV 120:REM	SETDEV wird zurückgesetzt (siehe unten)
130:LPRINT "Variab le (A\$):";A\$ 140:REM 150:END	hier wird nun die empfangene Variable auf dem CE 150 ausgedruckt

Zeile 110 : diese Zeile bedarf noch einer besonderen Aufmerksamkeit. Mit der Anweisung SETDEV werden die Parameter für SETDEV auf Null gesetzt. Sonst könnte kein Ausdruck auf dem CE 150 erfolgen, da der Rechner noch im Schnittstellenbetrieb arbeitet.

Programme vollständig von SHARP PC 1500(A)
zu SHARP PC 1500(A) mit CE 158 übertragen

Es ist möglich mit der CE 158 von SHARP gesamte Programme von einem zum anderem Rechner zu Übertragen. Demnach ist es sehr einfach auch Programme mit einem Akustikkoppler über das Telefonnetz einem Computerkollegen zuzusenden. Einen Anschlußplan für den richtigen "DRAHT" liegen den meisten Akustikkopplern bei. Darum möchte ich hier nicht weiter auf die Anschlußbelegung eingehen.

Zur Software für Datenfernübertragung (kurz DFÜ) mit SHARP PC 1500(A)

Für die Übertragung von Programmen von SHARP PC 1500(A) nach SHARP PC 1500(A) benötigt man keine spezielle Software wie zum Beispiel beim Übertragen von Variablen.

Sendereingabe

Erklärung

SETCOM 300,8,N,1 — für eine schneller Übertragung ev. höher Baud-Rate
SETDEV CO — siehe Tabelle unter CO
CSAVEa — Programm ohne Namen übertragen
CSAVE "Programmname" — Programm mit spezifischen Programmnamen übertragen
CSAVE M"Programmname";Startadresse,Endadresse — MAPROs übertragen

Empfängereingabe

Erklärung

SETCOM 300,8,N,1 — gleiche Parameter wie Sender
OUTSTAT 0 — Handshake vorbereiten
SETDEV CI — siehe Tabelle unter CI
CLOADa — Programm ohne Namen empfangen
CLOAD "Programmname" — Programm mit spezifischen Namen empfangen
MERGEa — Programm zu dem im Speicher dazuMERGEN
MERGE "Programmname"=Programm mit Namen dazuMERGEN
CLOAD M"Programmname"=MAPRO=empfangen (MAPRO=Maschinenprogramm)

Spezielle Anwendung mit EP 44(BROTHER).

Mit dem kleinem untenstehenden Programm im SHARP PC 1500(A) können Sie auf dem EP 44 (Tastatur) Daten (Variablen) eingeben, die dann auf dem LCD-Display zu Sehen sind.

Programmlisting EP 44 --- SHARP PC 1500(A)

Erklärung

```
10:"B"REM **EMPPFA
    ENGER EP 44->>
    PC 1500**
20:REM
30:OUTSTAT 0
40:REM
50:SETCOM 110,8,N
    ,1
60:REM
70:WAIT 0
80:REM
90:As=RINKEY$
100:REM
110:IF As=CHR$ 13
    THEN 150
120:REM
130:PRINT As;:GOTO
    70
140:REM
150:SETDEV
160:END
```

Start mit DEF B
Rechner wird für Datenempfang vorbereitet
SETCOM mit 110 Baud geht es am Besten
die empfangen Daten werden sofort vom PC 1500(A) in A\$ geladen SETDEV nicht nötig
wenn Sie RETURN auf der EP 44 endet das Programm
Normalzustand herstellen

- A.1. DIE SCHNITTSTELLE RS232C
- A.2. CHECKLISTE ZUR DATENÜBERTRAGUNG
- A.3. HÄUFIG BENUTZTE ZEICHENSÄTZE

DIE RS232C-SCHNITTSTELLE

A.1.

Die Norm RS232C ist eine amerikanische Industriennorm nach EIA (Electronic Industries Association). Die entsprechende internationale Norm ist nach CCITT (Comité Consultatif International Telephonique et Télégraphique) die Schnittstellenspezifikation V.24/V.28 (V.24 beschreibt die funktionellen, V.28 die elektrischen Eigenschaften). Die entsprechende deutsche Industriennorm ist DIN 66020/66022.

Die RS232C-Norm beschreibt eine asynchrone, bitserielle, voll duplex (d.h. Datenverkehr in beiden Richtungen gleichzeitig möglich) Schnittstelle, die anfangs für Modems gedacht war, heute jedoch häufig bei Computer-Peripherie-Verbindungen angewendet wird.

Diese serielle Schnittstelle läßt eine Datenübertragung mit einer Leitungslänge bis 30 Metern zu; die Länge ist jedoch umgekehrt proportional zur Übertragungsgeschwindigkeit (von 50 bis 19.200 Baud; ein Baud ist ein Bit/Sekunde).

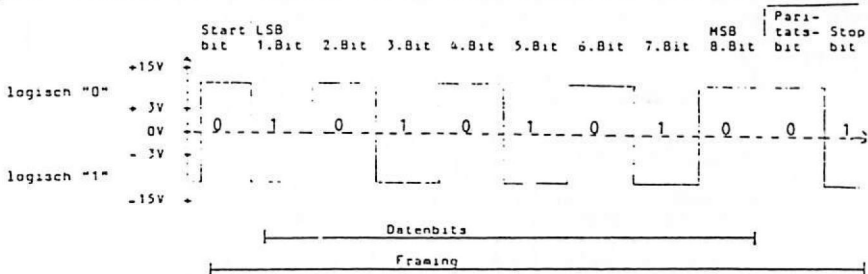
Die Daten werden in negativer Logik gesendet, d.h. ein H-Pegel (+3 V bis +15 V) ist logisch "0" (auch Space genannt) und ein L-Pegel (-3 V bis -15 V) ist logisch "1" (auch Mark genannt).

Ein Übertragungsrahmen (d.h. alle notwendigen Bits um ein Datenbyte zu senden), auch Framing genannt, wird eingeleitet durch ein Startbit mit H-Pegel (weil im "Ruhezustand" der L-Pegel anliegt) und abgeschlossen durch ein einhalb oder zwei Stopbits mit L-Pegel (um zu garantieren, daß der Pegel wieder in den "Ruhezustand" kommt).

Die Datenbits (wahlweise sieben oder acht Bits) und - wenn erwünscht - ein Paritätsbit (wahlweise gerade oder ungerade Parität) werden darin eingeschlossen.

Zum Beispiel würde das große "U" - nach der ASCII-Tabelle (hex: 55 / dez: 85) mit dem Format: acht Datenbits, gerade Parität und ein Stopbit - so aussehen:

ZENTRALSERVICE



Das ETX/ACK-Protokoll ist ebenfalls eine Software-Prozedur. Bei diesem Protokoll werden die ASCII-Zeichen ETX (hex: 03 / dez: 3) und ACK (hex: 06 / dez: 6) angewendet. Zu Beginn der Übertragung sendet das Empfangsgerät ACK-Zeichen um Daten anzufordern und die Übertragung einzuleiten. Das Sendegerät beginnt daraufhin ein Datenpaket, das mit dem ETX-Zeichen abgeschlossen wird zu senden und geht in den Wartezustand über. Dieses Datenpaket wird vom Empfänger sichergestellt und durch erneutes Senden des ACK-Zeichens ein neues Datenpaket angefordert. Hierbei ist zu beachten, daß die Geräte immer nur eine bestimmte Zeit warten, und daß das Datenpaket nicht die Pufferkapazität des Empfangsgerätes überschreitet.

Das Framing kann also eine Länge von 9 bis 12 Bits haben. Mit höherer Bit-Anzahl steigt die Datenübertragungssicherheit, jedoch sinkt der Informationsgehalt pro Frame, d.h. das Verhältnis von Framing zu Datenbits wird schlechter.

Wenn man nun die Schnittstellenparameter festgelegt hat ((Baud-Rate in Abhängigkeit von der Kabellänge), Anzahl der Datenbits, Parität, Anzahl der Stopbits), ist noch ein sehr wichtiger Punkt das Übertragungsprotokoll, welches die Verständigungsart zwischen zwei Einheiten festsetzt. Die meist benutzten sind das Hardware-, X-on/X-off und das ETX/ACK-Protokoll.

Das Hardware-Protokoll ist am weitverbreitetsten, da keine spezielle Software für die ordnungsgemäße Kommunikation zweier Geräte benötigt wird. Zur Signalisierung des Zustandes "bereit" (kann Daten empfangen) oder "beschäftigt" (kann zur Zeit keine Daten empfangen) wird im einfachsten Fall nur eine Leitung benötigt, nämlich Pin 4 = "Request to Send" (häufig auch Pin 20 = DTR). Ist diese Leitung (Handshake-Leitung) positiv, so besteht Empfangsbereitschaft, ist sie hingegen negativ, signalisiert sie den "beschäftigt"-Zustand.

Das X-on/X-off-Protokoll ist eine rein softwaremäßige Prozedur, die meistens schon fest im Gerät implementiert ist, anderenfalls muß der Benutzer diese Prozedur programmtechnisch einfügen. Dieses Protokoll arbeitet mit zwei Steuerzeichen X-on (ASCII-Zeichen DC1, dies entspricht hex: 11 / dez: 17) und X-off (ASCII-Zeichen DC3, dies entspricht hex: 13 / dez: 19). Das Gerät sendet bei Empfangsbereitschaft ein X-on Zeichen und im anderen Fall ("beschäftigt"-Zustand) ein X-off Zeichen.

Nachdem man nun die Schnittstellenparameter und das Übertragungsprotokoll bei beiden Geräten eingestellt hat, ist eine einwandfreie Kommunikation erst möglich, wenn folgende Randbedingungen erfüllt sind:

- die Geräte den gleichen Zeichensatz benutzen (ASCII, IBM, ...)
- den gleichen Zeilenendcode benutzen (ASCII-Zeichen: C_R L_F, C_R, ...)
- den gleichen Textendcode (z.B. ASCII-Zeichen: Subnach Beendigung der Übertragung austauschen)
- in der gesendeten Datei keine SteuerCodes versteckt sind, deshalb sollte man Programme nur im Klartext (ASCII-Datei) senden

Da in sehr vielen Fällen von der Norm abgewichen wird, ist eine Verbindung von zwei Geräten über die RS232C Schnittstelle oft nur nach einer näheren Untersuchung des Schnittstellenverhaltens dieser beiden Geräte möglich.

Die Standard-Beschaltung von Computer und Peripherie (DTE) mit den am häufigsten vorkommenden Signalleitungen:

<u>Peripherie:</u>		<u>Computer:</u>			
engl. Abkürzung	Pin Nr.	engl. Abk.	deut. Bezeichnung	Zweck/Anwendung	
FG	----- 1	FG	Schutzerde	Hat Schutzfunktion für die Signalübertragung. Ist angeschlossen an Kabelabschirmung, Gerätegehäuse, Netzerde.	
RD	----- 2	TD	Sendedaten	Serielle Bit Übertragung (Sendung).	
TD	----- 3	RD	Empfangsdaten	Serieller Bit Empfang	
CTS	----- 4	RTS	Sendeteil einschalten	Zeigt an, ob bereit ist Daten auszutauschen (Handshake-Leitung).	
RTS	----- 5	CTS	Sendebereitschaft	Überprüft ob Peripherie bereit ist, Daten auszutauschen.	
DTR	----- 6	DSR	Betriebsbereitschaft	Überprüft, ob die SIO der Peripherie eingeschaltet ist.	
SG	----- 7	SG	Betriebserde	"Nullpotential" der Signale (meistens mit FG verbunden).	
"Ready" (+15 V)	----- 8	DCD	Empfangssignalpegel	Überprüft, ob die Peripherie überhaupt eingeschaltet ist.	
DSR	----- 20	DTR	Terminal betriebsbereit	Zeigt an, ob SIO eingeschaltet ist.	

	Kurzzzeichen			Stecker- belegung	Beschreibung	
	CCITT V.24	EIA RS 232	DIN 66020			
Daten Erde	101	AA	E 1	1 ---	Schutzerde	Protective ground
	102	AB	E 2	7 ---	Signalerde/Betriebserde	Signal ground/Common return
Daten Erde	103	BA	D 1	2 A	Sendedaten	Transmitted data (TD)
	104	BB	D 2	3 E	Empfangsdaten	Received data (RD)
Steuer- und Meldesignale	105	CA	S 2	4 A	Sendeteil einschalten	Request to send (RTS)
	106	CB	M 2	5 E	Sendebereitschaft	Clear to send (CTS)
	107	CC	M 1	6 E	Betriebsbereitschaft	Data set ready (DSR)
	108.1		S 1.1	20 A	Übertragungsleitung anschalten	Connect data set to line
	108.2	CD	S 1.2	20 A	Terminal betriebsbereit	Data terminal ready (DTR)
	125	CE	M 3	22 E	Ankommender Ruf	Ring indicator (RI)
	109	CF	M 5	8 E	Empfangssignalpegel	Received line signal detector (RLSD)
	110	CG	M 6	21 E	Empfangsgüte	Signal quality detector (SQD)
	111	CH	S 4	23 A	Übertragungsgeschwindigkeit (Wahl vom Terminal)	Data signal rate selector (DSRS)
112	CI	M 4	23 E	Übertragungsgeschwindigkeit (Wahl vom Modem)	Data signal rate selector (DCE)	
126	CK	S 5	11 A	Wahl Sendefrequenz (200 baud Modem)	Select transmit frequency (200 baud modem)	
Takte	113	DA	T 1	24 A	Sendeschrifttakt von DEE	Transmitter signal element timing (Transmit clock to modem DTE)
	114	DB	T 2	15 E	Sendeschrifttakt von DUE	Transmitter signal element timing (TC) (Transmit clock from modem DCE)
	115	DD	T 4	17 E	Empfangsschrifttakt	Receiver signal element timing (RC) Receive clock
Frei Zusatzkanal	118	SBA	HD 1	14 A	Sendedaten Rückkanal	Secondary transmitted data
	119	SBB	HD 2	16 E	Empfangsdaten Rückkanal	Secondary received data
	120	SCA	HS 2	19 A	Rückkanal Sendeteil einschalten	Secondary request to send
	121	SCB	HM 2	13 E	Rückkanal Sendebereitschaft	Secondary clear to send
	122	SCF	HM 5	12 E	Rückkanal Empfangssignalpegel	Secondary Carrier detector
			9/10 11/18/25	Zur Verwendung für Prüfgeräte Nicht belegt	Reserved for data set testing Unsigned	

CHECKLISTE ZUR DATENÜBERTRAGUNG

A.2.

Damit ein einwandfreier Datenaustausch stattfinden kann, sind folgende Punkte bei der Kommunikation über die RS232C Schnittstelle zu beachten:

- das richtige Kabel anschließen
- die gleichen Schnittstellenparameter setzen (Übertragungsrate, Parität, Anzahl der Daten- und Stopbits)
- für das gleiche Übertragungsprotokoll (- das Handshaking muß in beiden Richtungen einwandfrei funktionieren) muß gesorgt sein (z.B.: Hardware-, X-on/X-off- oder ETX/ACK-Protokoll)
- die richtigen Kanäle (z.B.: Printer-Kanal, Tastatur, Floppy...) auf die RS232C Schnittstelle leiten
- der Zeichensatz muß gleich sein (z.B.: IBM, ASCII, Deutsch ASCII-Referenz-Version,...)
- der Zeilenendcode muß gleich sein (z.B.: C_R L_F, C_R,...)
- der Textendcode muß gleich sein (z.B.: S_{ub},...)
- Besonderheiten der jeweiligen Geräte müssen berücksichtigt werden (z.B.: "Device Time Out"-Zeiten, ...).

MEHR FREIHEIT FÜR

NEUES DENKEN:

SHARP POCKET COMPUTER.

HÄUFIG BENUTZTE ZEICHENSÄTZE

HÄUFIG BENUTZTE ZEICHENSÄTZE

. 3 .

Deutscher ASCII-Zeichensatz

Upper Bit Lower Bit	Hex.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hex.	Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	NUL		SP	!	@	A	B	C								
		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	0001		!	"	#	\$	%	&	'	()	*	+	,	-	.	:
		1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	0010		DC2	"	2	B	R	b	r								
		2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	0011			#	3	C	S	c	s								
		3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	0100		DC4	\$	4	D	T	d	t								
		4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	0101			"	5	E	U	e	u								
		5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	0110			&	6	F	V	f	v								
		6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	0111		DEL		7	G	W	g	w								
		7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	1000		HS	CAN	(X	H	X	h	x							
		8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	1001		HT)	9	I	Y	i	y								
		9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A	1010		LF	*	:	J	Z	j	z								
		10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	1011		VT	LSC	+	:	K	A	k	a							
		11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C	1100			.	<	L	O	l	o								
		12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D	1101		CR	-	=	M	U	m	u								
		13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E	1110		SO		>	N	.	n	.								
		14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F	1111		SI	/	'	O	--	o	DEL								
		15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

FISCHER GMBH -

US-ASCII Standard-Zeichensatz

Upper Bit Lower Bit	Hex.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hex.	Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	NU	SP	!	"	#	\$	%	&	'	()	*	+	,	;	:
1	0001			!	"	#	\$	%	&	'	()	*	+	,	;	:
2	0010	DC2	"	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3	0011			#	\$	%	&	'	()	*	+	,	;	:	:	:
4	0100	DC4	\$	%	&	'	()	*	+	,	;	:	:	:	:	:
5	0101		"	5	6	7	8	9	U	e	u						
6	0110		&	6	7	8	9	F	V	f	v						
7	0111	DEL		7	8	9	G	W	g	w							
8	1000	BS	CAN	(8	9	H	X	h	x							
9	1001	HT)	9	I	Y	i	y									
A	1010	LJ	*	:	J	Z	j	z									
B	1011	VT	ESC	+	:	K	[]	k]							
C	1100	FF		<	[]	;	:									
D	1101	CR	-	=	M	[]	~	m	~							
E	1110	SO		>	N]	^	~	n	~							
F	1111	SI	/	?	O	^	~	DEL									

IBM-1 Zeichensatz

Upper Bit	Lower Bit	Hex.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Hex.	Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	NUL	SP	0	1	A	P	~	p			ä	⋮	⌂	⌂	α	≡	
			0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	0001		!	@	#	\$	%	&	'	()	*	+	,	-	.	/	
			1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	0010	DC2	"	2	B	R	h	r		DC2	ó	⊗	⊗	T	T	Γ	≥	
			2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	0011		#	3	C	S	c	s			ü			†	†	π	≤	
			3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	0100	DC4	\$	4	D	T	d	t		DC4	ñ	†	†	†	†	Σ	∩	
			4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	0101		%	5	E	U	e	u			Ñ	†	†	†	†	σ	J	
			5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	0110		&	6	F	V	f	v			⊗	†	†	†	†	μ	+	
			6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	0111	BFL		7	G	W	w	BFL			⊗	†	†	†	†	τ	≈	
			7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	1000	BS	CAN	8	H	X	h	x	BS	CAN	⊗	†	†	†	†	ϕ	∩	
			8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	1001	HT		9	I	Y	i	y	HT		†	†	†	†	†	∩	•	
			9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A	1010	LF	*		J	Z	j	z	LF		†	†	†	†	†	Ω	•	
			10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	1011	VT	LSC		.	K		k	VT	FSC	1/2	†	†	†	†	δ	√	
			11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C	1100	FF			<	L	\	l	;	FF	1/4	†	†	†	†	∞	∩	
			12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D	1101	CR	-	=	M		m	}	CR		†	†	†	†	†	∩	∩	
			13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E	1110	SO	>	N	-	n	-	SO			†	†	†	†	†	€	■	
			14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F	1111	SI	/	∩	O	_	∩	SI		»	†	†	†	†	†	∩	SP	
			15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

FISCHER GMBH

SHARP

Do not sale!

IBM-2 Zeichensatz

Lower Bit \ Upper Bit	Hex.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Hex.	Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110
0	0000		NUL	SP	0	1	P	q	r	s	t	u	v	w	x	y	z
			0	16	32	48	64	80	96	112	128	144	160	176	192	208	224
1	0001		⊕		!	A	O	a	q	u	x	i	⊕	⊕	⊕	⊕	⊕
			1	17	33	49	65	81	97	113	129	145	161	177	193	209	225
2	0010		⊕	DC2	2	B	R	h	r	e	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			2	18	34	50	66	82	98	114	130	146	162	178	194	210	226
3	0011		♥		#	C	S	e	s	ä	ö	u	!	!	!	!	!
			3	19	35	51	67	83	99	115	131	147	163	179	195	211	227
4	0100		⊕	DC4	4	D	T	J	a	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			4	20	36	52	68	84	100	116	132	148	164	180	196	212	228
5	0101		⊕	8	5	F	U	e	u	a	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			5	21	37	53	69	85	101	117	133	149	165	181	197	213	229
6	0110		⊕	⊕	6	F	V	I	v	a	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			6	22	38	54	70	86	102	118	134	150	166	182	198	214	230
7	0111		BLL	⊕	7	G	W	g	w	ç	u	⊕	⊕	⊕	⊕	⊕	⊕
			7	23	39	55	71	87	103	119	135	151	167	183	199	215	231
8	1000		BS	CAN	⊕	X	H	X	h	x	e	y	⊕	⊕	⊕	⊕	⊕
			8	24	40	56	72	88	104	120	136	152	168	184	200	216	232
9	1001		HT	⊕	9	I	Y	i	y	e	O	⊕	⊕	⊕	⊕	⊕	⊕
			9	25	41	57	73	89	105	121	137	153	169	185	201	217	233
A	1010		LF	→	*	J	Z	j	z	e	U	⊕	⊕	⊕	⊕	⊕	⊕
			10	26	42	58	74	90	106	122	138	154	170	186	202	218	234
B	1011		VT	ESC	+	K	I	k	{	i	e	⊕	⊕	⊕	⊕	⊕	⊕
			11	27	43	59	75	91	107	123	139	155	171	187	203	219	235
C	1100		FF		<	L	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			12	28	44	60	76	92	108	124	140	156	172	188	204	220	236
D	1101		CR		=	M		m		v	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			13	29	45	61	77	93	109	125	141	157	173	189	205	221	237
E	1110		SO		>	N	-	n	-	A	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			14	30	46	62	78	94	110	126	142	158	174	190	206	222	238
F	1111		SI		/	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
			15	31	47	63	79	95	111	127	143	159	175	191	207	223	239

**Das Beste für Wirtschaft,
Wissenschaft und Technik.**

Service Information

Ref. Nr.:

Nr. : RS-025

Modell : PC-1600

Datum :

Betr.:

Beschreibung der Seriellen-Schnittstelle für den PC-1600
Pocket Computer

1. Hardware:

a)		RS232C		Pin	E/A	1	2	3
Bez.	Standardbez.							
SD	TD			2	A			+/-
RD	KD			3	E			
RS	RTS			4	A	-	+od.-	+
* CS	CTS			5	E			
DR	DSR			6	E			
SG	SG			7	-			
CD	DCD			8	E			
CI	RI			9	E			
Vcc	-			10	A	+	+	+
ER	DTR			14	A	-	+od.-	+

1= nach dem Einschalten

2= je nachdem, wie die Parameter gesetzt wurden (OUTSTAT,..)

3= beim Senden (bzw. Empfangen) beeinflussbar mit den Befehlen RCVSTAT und SNDSTAT).

*= Status-Abfrage:

Beim Senden und Empfangen wird nur CS abgefragt.

(Beeinflussbar mit den Befehlen RCVSTAT und SNDSTAT).

+= entspricht einem High Pegel im Bereich von +3 V bis +15 V.

-= entspricht einem Low Pegel im Bereich von -3 V bis -15 V.

+/-= entspricht einem Datenverkehr mit wechselndem Low- und High Pegel

b) Benötigter Stecker: Spezial Stecker (CE-1605 L).

c) Bemerkung:

an Pin 10 (Vcc) liegt ein Pegel von +5 V an, ansonsten ist der Pegel ±8 V.

2. Protokolle:

Der PC-1600 empfängt das Hardware-Protokoll, sendet aber keins.

Ein x-on/x-off Protokoll kann der PC-1600 ausführen, wenn dies bei SETCOM gesetzt worden ist.

3. Betriebssystem: Sharp Basic

a) Senden / Empfangen:

Parameter setzen:

```
SETCOM "COM1:", 1200, 8, N, 1, X
OUTSTAT "COM1:", 0
RCVSTAT "COM1.", 63
SNDSTAT "COM1:", 59
INIT "C1M1:", "", ""
```

Das Programm wird zuerst in den Arbeitsspeicher gebracht (z.B.: LOAD"X:Programm") und dann mit

```
SAVE "COM1:", A
```

gesendet.

Programme werden empfangen, indem die Parameter wie oben, bis auf

```
INIT"COM1:", 600, "", ""
```

gesetzt werden.

Dann wird mit

```
LOAD"COM1:"
```

das Programm in den Arbeitsspeicher geladen.

b) Mit den Dateibefehlen (Kennzeichen #) wird analog zu 3. c) gearbeitet.

c) Es ist auch möglich über den Printer Kanal zu senden:

Die Parameter werden wie bei a) gesetzt und zusätzlich

```
PCONSULE"COM1:", 80, 2 und
```

```
SETDEV"COM1:", Po
```

einggegeben.

Nun kann man mit:

```
LLIST, LPRINT oder
LFILES
```

senden, wobei man am Ende der Übertragung noch einmal

```
LPRINT CHR$(26)
```

eingeben muß.

d) Der Textencode entspricht dem ASCII-Code S_{ub} (bzw. dez.: 26/hex.:1A).

Der Zeilenencode ist $C_R L_F$ (bzw. dez.: 13 u. 10/hex.: 0D u. 0A).

Für den Printer Kanal ist in unserem Beispiel mit dem Befehl "PCONSULE" dieser auf $C_R L_F$ gesetzt worden (möglich ist auch: nur C_R und nur L_F).

- e) Benutzer Zeichensatz:
Siehe Benutzer Handbuch (= ASCII).
- f) Bemerkungen:
Die maximale Übertragungsgeschwindigkeit bei den Befehlen:
INPUT; INPUT#; PRINT#;
LIST und LPRINT
beträgt 4.800 Baud
(sonst ist auch 9.600 Baud möglich)

Es sind Kabelverbindungen vorhanden für:

- PC-7100/7000 Computersystem, Bez.: CE-1604L
(+ Programme in GW-Basic) siehe RS-011
- PC-5000 Computersystem, Bez.: CE-1603L
(+ Programme in GW-Basic)
- MZ-5600 Computersystem, Bez.: CE-1602L
(+ Programme in GW-Basic) siehe RS-017
- Modem, Bez.: CE-1601L
- nur Stecker mit Kabel, Bez.: CE-1605L
- und Spezifikationen für:
PC-7500 Computersystem
(+ Programme in GW-Basic) siehe RS-004

ASCII-Code Tabelle PC-1600

Zeichen Codes im MODE Ø

Im MODE Ø schließt der Zeichensatz des PC-1600 internationale Sonderzeichen, grafische Symbole und griechische Schriftzeichen ein, die zusätzlich zu den üblichen Buchstaben, Ziffern, Satzzeichen und sonstigen ASCII-Symbolen dargestellt werden können. Dieser Zeichensatz entspricht dabei genau dem eines IBM-PC.

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL															
1																
2																
3																
4																
5																
6																
7	BEL															
8	BS															
9	HT															
A	LF															
B	VT															
C	FF															
D	CR															
E	SO															
F	SI															

FISCHER GMBH -

Zeichen Codes in MODE 1

In MODE 1 wird der Zeichensatz des PC-1600 modifiziert, damit er Kompatibilität zum PC-1500 gewähren kann. Die folgende Tabelle zeigt, welche Unterschiede in den beiden Anzeige-Modi bestehen:

HEX CODE	27	5B	5C	5D	5E	5F	60	7B	7C	7D	7E	7F
MODE 0	'	[\]	~	_	'	(:	}	~	⏏
MODE 1		√	≠	∏	^						~	

Service Information

Ref. Nr.:
 Nr. : RS-026
 Modell : PC-1500 A
 Datum :

Betr.:

Beschreibung der Seriellen-Schnittstelle für den PC-1500 A mit CE-158 Pocket-Computer

1. Hardware:

a)	RS232C					
	Bez.	Standardbez.	Pin	E/A	1	2
	TD	TD	2	A	-	+/-
	RD	RD	3	E		
	RTS	RTS	4	A	-	+od.-
*	CTS	CTS	5	E		
(*)	DSR	DSR	6	E		
	SG	SG	7	-		
(*)	CD	DCD	8	E		
	DTR	DTR	20	A	-	+od.-

1= nach dem Einschalten

2= je nachdem wie OUTSTAT gesetzt worden ist
 (wenn nicht gesendet wird, ist Pin 2 negativ)

*= Status-Abfrage:

Wenn CD oder DSR "low" ist, kann nicht gesendet oder empfangen werden, aber es können diese Leistungen "offen" gelassen werden. Ansonsten wird nur CTS abgefragt.

+= entspricht einem High Pegel im Bereich von +3 V bis +15 V.

-= entspricht einem Low Pegel im Bereich von -3 V bis -15 V.

+/-= entspricht einem Datenverkehr mit wechselndem Low- und High Pegel.

b) Benötigter Stecker: D-25-Pin.

- c) Bemerkung:
Der Pegel beträgt etwa $\pm 8,5$ V.

2. Protokolle:

Der PC-1500 A + CE-158 empfängt das Hardware-Protokoll, sendet aber keins.

Das x-on/x-off Protokoll ist wahrscheinlich im Terminalmodus (bzw. DTE-Modus) möglich, dies bleibt jedoch näher zu untersuchen.

3. Betriebssystem: Sharp Basic

a) Senden / Empfangen

Parameter setzen:

z. B. OUTSTAT Ø
SETCOM 12ØØ, 8, N, 1
SETDEV PO, CI, CO

Ein Programm wird aus dem Arbeitsspeicher gesendet mit:
CSAVEa

und empfangen mit:
LOADa

- b) Mit den Dateibefehlen (Kennzeichen #) wird analog zu 3.c) gearbeitet.

- c) Es ist auch möglich über den Printer Kanal zu senden:

Parameter setzen:

z.B.: OUTSTAT Ø
SETCOM 12ØØ, 8, N, 1
CONSOLE 8Ø,Ø
SETDEV PO, CI, CO

Nun kann man mit:

LLIST oder LPRINT
senden, wobei man am Ende der Übertragung noch einmal
LPRINT CHR\$(13)
eingeben muß.

- d) Der Textendcode entspricht dem ASCII-Code $C_R C_R$ (zweimal dez.: 13/hex.: ØD).

Der Zeilenendcode ist C_R . Mit CONSOLE wird nur für den Printer Kanal der Zeilenendcode gesetzt (im Beispiel bei c) ist es C_R ; möglich ist auch:
 $L_F, C_R L_F, C_R C_R, L_F C_R, L_F L_F$:

Mit den Dateibefehlen (siehe 3. b) kann man den Text- und Zeilenendcode beliebig festsetzen.

- e) Benutzer Zeichensatz:
Siehe folgende Seite.

Spezifizierte Kabelverbindungen (+ Programme in GW-Basic)
gibt es für:

- PC-7500 Computersystem unter RS-005
- PC-7100/7000 Computersystem unter RS-012
- MZ-5600 Computersystem unter RS-016

ASCII-Code Tabelle PC-1500 A

ASCII Zeichen	Entsprechender Code		
	Binär	Dzimal	Hexadezimal
space	00100000	32	20
!	00100001	33	21
"	00100010	34	22
#	00100011	35	23
\$	00100100	36	24
%	00100101	37	25
&	00100110	38	26
'	00100111	39	27
(00101000	40	28
)	00101001	41	29
*	00101010	42	2A
+	00101011	43	2B
,	00101100	44	2C
-	00101101	45	2D
.	00101110	46	2E
/	00101111	47	2F
0	00110000	48	30
1	00110001	49	31
2	00110010	50	32
3	00110011	51	33
4	00110100	52	34
5	00110101	53	35
6	00110110	54	36
7	00110111	55	37
8	00111000	56	38
9	00111001	57	39
:	00111010	58	3A
;	00111011	59	3B
<	00111100	60	3C
=	00111101	61	3D
>	00111110	62	3E
?	00111111	63	3F
@	01000000	64	40
A	01000001	65	41
B	01000010	66	42
C	01000011	67	43
D	01000100	68	44
E	01000101	69	45
F	01000110	70	46
G	01000111	71	47
H	01001000	72	48

ASCII Zeichen	Entsprechender Code		
	Binär	Dzimal	Hexadezimal
P	01010000	80	50
Q	01010001	81	51
R	01010010	82	52
S	01010011	83	53
T	01010100	84	54
U	01010101	85	55
V	01010110	86	56
W	01010111	87	57
X	01011000	88	58
Y	01011001	89	59
Z	01011010	90	5A
[01011011	91	5B
\	01011100	92	5C
]	01011101	93	5D
^	01011110	94	5E
_	01011111	95	5F
	01100000	96	60
a	01100001	97	61
b	01100010	98	62
c	01100011	99	63
d	01100100	100	64
e	01100101	101	65
f	01100110	102	66
g	01100111	103	67
h	01101000	104	68
i	01101001	105	69
j	01101010	106	6A
k	01101011	107	6B
l	01101100	108	6C
m	01101101	109	6D
n	01101110	110	6E
o	01101111	111	6F
p	01110000	112	70
q	01110001	113	71
r	01110010	114	72
s	01110011	115	73
t	01110100	116	74
u	01110101	117	75
v	01110110	118	76
w	01110111	119	77
x	01111000	120	78

Professioneller Einsatz als Ideal.

SHARP SHARP

ALLES FÜR SHARP-COMPUTER

I	01001001	73	49	y	01111001	121	79
J	01001010	74	4A	z	01111010	122	7A
K	01001011	75	4B	{	01111011	123	7B
L	01001100	76	4C	:	01111100	124	7C
M	01001101	77	4D	}	01111101	125	7D
N	01001110	78	4E	~	01111110	126	7E
O	01001111	79	4F	■	01111111	127	7F

Service Information

Ref. Nr.:

Nr. : RS-027

Modell : PC-1360

Datum :

Betr.:

Beschreibung der Seriellen-Schnittstelle für den PC-1360 mit CE-130 T Pocket-Computer

1. Hardware:

a)	Bez.	RS232C Standardbez.	Pin	E/A	1	2
	SD	TD	2	A	-	+/-
	RD	RD	3	E		
	RS	RTS	4	A	-	+
*	CS	CTS	5	E		
	DR	DSR	6	E		
	SG	SC	7	-		
*	CD	DCD	8	E		
	RR	-	11	A	-	-
	ER	DTR	20	A	+	+

Nach dem Einschalten sind an den Pins nur geringe Spannungen ($< \pm 1$ V).

1= nach dem Öffnen des "Kanals" mit OPEN und beim Empfangen

2= beim Senden

*= Status-Abfrage:

Beim Empfangen wird nur CD und beim Senden nur CS abgefragt.

+= entspricht einem High Pegel von +3 V bis +15 V.

-= entspricht einem Low Pegel von -3 V bis -15 V.

+/-= entspricht einem Datenverkehr mit wechselndem Low- und High Pegel.

b) Benötigter Stecker: D-25-Pin

c) Bemerkung:

Der Pegel beträgt etwa ± 8 V.

2. Protokolle:

Der PC-1360 + CE-130 I empfängt das Hardware-Protokoll, sendet aber keins.

Das x-on/x-cff Protokoll ist nicht möglich, (nur mit der Übertragungsform 3.b mit entsprechender Software).

3. Betriebssystem: Sharp Basic

a) Senden / Empfangen:

Parameter setzen mit
z.B.: OPEN "1200, N, 8, 1, A, C, &1A"
CONSOLE 80

Ein Programm wird aus dem Arbeitsspeicher gesendet mit:
SAVE

und empfangen mit:
LOAD

b) Mit den Dateibefehlen (Kennzeichen #) wird analog zu 3.c) gearbeitet.

c) Es ist auch möglich über den Printer Kanal zu senden mit:
LLIST und LPRINT
wenn wie bei a) die Parameter vorher gesetzt wurden, wobei man am Ende der Übertragung noch einmal
LPRINT CHR\$(26)
eingeben muß.

d) Der Textendcode kann mit dem OPEN Befehl beliebig gesetzt werden; in diesem Fall entspricht es dem ASCII-Code: Sub (bzw. dez.:26/hex.:1A).

Der Zeilenendcode ist mit dem OPEN Befehl auf C_R gesetzt worden (möglich ist auch: C_RL_F und L_F)

Mit dem Dateibefehlen (siehe 3.b) kann man den Text- und Zeilenendcode beliebig festsetzen.

e) Benutzer Zeichensatz:
Siehe Benutzer Handbuch (=ASCII)

Bemerkung:
Ein Handshaking wird in den Beispielen a) und c) leider nicht ausgeführt; mit entsprechender Software über Beispiel b) läßt es sich jedoch realisieren.
Der vielseitige OPEN-Befehl erleichtert das Anpassen an Fremdgeräte.

Spezifizierte Kabelverbindungen (+ Programme in GW-Basic) gibt es für:

- PC-7500 Computersystem unter RS-006
- PC-7100/7000 Computersystem unter RS-013
- MZ-5600 Computersystem unter RS-019

ALLES FÜR SHARP-COMPUTER

ASCII-Code Tabelle PC-1360

Ersten 4 Bits

Hex. /Binar	0	1	2	3	4	5	6	7	8	E	F
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1110	1111
0	0	16	32	48	64	80	96	112	128	224	240
0000	NUL		SPACE	0	@	P	t	p			
1	1	17	33	49	65	81	97	113	129	225	241
0001		!	1	A	Q	a	q				
2	2	18	34	50	66	82	98	114	130	226	242
0010		"	2	B	R	b	r				
3	3	19	35	51	67	83	99	115	131	227	243
0011		#	3	C	S	c	s				
4	4	20	36	52	68	84	100	116	132	228	244
0100		\$	4	D	T	d	t				
5	5	21	37	53	69	85	101	117	133	229	245
0101		%	5	E	U	e	u				♦
6	6	22	38	54	70	86	102	118	134	230	246
0110		&	6	F	V	f	v				♥
7	7	23	39	55	71	87	103	119	135	231	247
0111		'	7	G	W	g	w				♦
8	8	24	40	56	72	88	104	120	136	232	248
1000		(8	H	X	h	x				♣
9	9	25	41	57	73	89	105	121	137	233	249
1001)	9	I	Y	i	y				■
A	10	26	42	58	74	90	106	122	138	234	250
1010		*		J	Z	j	z				▬
B	11	27	43	59	75	91	107	123	139	235	251
1011		+		K		k	l				⋄
C	12	28	44	60	76	92	108	124	140	236	252
1100		,		<	L		l				√
D	13	29	45	61	77	93	109	125	141	237	253
1101		-		*	M		m	l			
E	14	30	46	62	78	94	110	126	142	238	254
1110		.		>	N	^	n	~			
F	15	31	47	63	79	95	111	127	143	239	255
1111		/		?	O	-	o				

Zweiten 4 Bits

 * Mit freundlicher Genehmigung von: *
 * ANDREAS FISCHER *
 * - Kundendienstleiter Informationssysteme - *
 * Zentralservice SHARP ELECTRONICS HAMBURG *

Do not sale!

ALLES FÜR SHARP-COMPUTER

MC-1500

Centronics-Ausgabe aus Macro:

Folgendes kleines Programm ermöglicht die Ausgabe von Texten oder einzelnen Zeichen über die Centronics-Schnittstelle des CE-158 aus Maschinenprogrammen. Das Programm ist in drei Teile gegliedert:

- 1) Die Initialisierung der Schnittstelle. (Hier CALL &5002)
- 2) Die Ausgabe eines Zeichens in A. (Hier CALL &5004)
- 3) Die Ausgabe von Texten. Diese Startadresse des Textes muß in X übergeben werden, die maximale Länge in A. Der Text kann aber auch schon vorzeitig mit 0 enden. (Hier CALL &5000. Dieser Call ist so ausgelegt, daß er auch aus BASIC erfolgen kann: CALL &5000,A\$)

Bernd Rüter

```

10 "START &5400, &54FF, &5500, &55FF;Speicher fuer Symboltabelle und Objektcode
11 "CALL &D02B ;loescht INPUT-Buffer (Inhalt &7BB0-&7BFF = &0D)
12 "CALL (&CC) ;Schreibt Pointer aus Systembereich in X-Register
13 "BYTE &69 ;mit Datenbyte &69 wird der HERGEpointer aus (&7B69/A) geladen
14 "LD Y,X ;Setzt Y=X : Register zeigen auf H-Byte 1. Zeilennummer
15 "HB LOI A, (Y) ;Wenn H-Byte der Znr = &FF ist Ende des Programms erreicht
16 "INC A ;A<--A+1: Z-Flag wird gesetzt, wenn A = &FF war
17 "JR Z, ANZEIGE ;dann Anzeige und Ruecksprung ins BASIC
18 "INC Y: INC Y ;Y zeigt auf Beginn Befehlscode in der aktuellen Zeile
19 "AN CALL (&22);Laden naechster Instruktion in U (<STRING> ueberspringen!)
20 "ZE CP UL, &D ;Pruefung ob Zeilenende erreicht ist (Z-Flag gesetzt)
21 "JR Z, HB ;dann zeigt Y auf H-Byte naechste Zeile
22 "SY CALL (&C4);Syntaxpruefung, welche Instruktion nach U geladen wurde
23 "WORD &F1AB ;Token fuer REN
24 "BYTE &24 ;JR wenn keine Uebereinstimmung mit REM
25 "CALL (&20) ;Setzt Y auf H-Byte naechste Zeile (REM-Zeile uebersprungen)
26 "JR HB
27 ";*****HEX-Zahlen &xxxx werden uebersprungen:
28 "CALL (&C4) ;Syntaxpruefung, ob HEX-zahl mit & folgt
29 "BYTE &26 ;ASCII von &
30 "BYTE &10 ;wenn UL (<) & Sprung nach WEITER
31 "HX CALL (&D2);Syntaxpruefung (N A C H Laden naechstes Zeichen)
32 "WORD &3039 ;in UL Ziffer 1...9?
33 "BYTE &02 ;Sprung wenn keine Ziffer
34 "JR HX ;naechstes Zeichen pruefen
35 "CALL (&D0) ;Syntaxpruefung (O H N E Laden naechstes Zeichen)
36 "WORD &4146 ;in UL Buchstabe A...F?
37 "BYTE &02 ;Sprung wenn kein Buchstabe A...F (Ende HEX-Zahl)
38 "JR HX ;naechstes Zeichen pruefen
39 "JR ZE
40 "WEITER
41 "PUSH U ;U = 1. Zeichen Variablenname ODER Token/anderes Zeichen
42 "CALL (&C6) ;Vorbereitung Syntaxpruefung
43 "CALL (&CE) ;ermittelt Variablennamen
44 "BYTE &A2 ;Steuerbyte - legt Variablentyp fest (siehe Text)
45 "BYTE &02 ;JR wenn Fehler

```

```

46 "JR SP      ;in U stand ein zulaessiger Variablenname (NUR 1. Zeichen!!)
47 "POP U     ;nach PUSH folgt POP - sonst laeuft Stack ueber!!
48 "JR AN     ;Y zeigt auf Beginn naechster Instruktion
49 "SP POP U  ;U enthaelt das 1. Zeichen einer Variablen (ASCII A ... Z)
50 "LD X,&7B6F:LD A,UL:ADD X,A:LD(X),A;Zeichen wird im INPUT-Buffer gespeichert
51 "JR AN     ;Suche nach naechster Variablen
52 "ANZEIGE   ;Anzeige der im Programm benuetzten Variablen A-Z bzw. A$-Z$
53 "LD U,&7BB0 ;Beginn <TEXT> in U (Anfang INPUT-Buffer)
54 "LD XL,&1A  ;Laenge <TEXT> in XL
55 "CALL &ED38 ;Anzeige <TEXT> in LCD (Beginn in U / Laenge in XL ab Spalte 8)
56 "CALL &D828 ;loescht INPUT-Buffer
57 "CALL (&46) ;Ruecksprung ins BASIC ohne Loeschen der Anzeige
58 "END

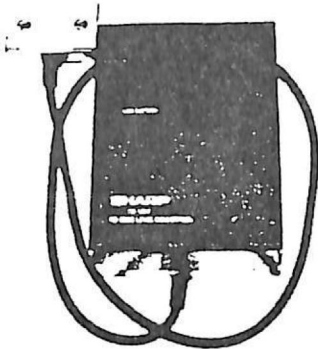
100 "START %5100,%5200,%5000,%50FF
110 " JR LOOP
120 " JR INIT
130 " JR COUT
140 "LOOP PUSH A:LDI A,(X):JR Z,ENDE
150 " CALL COUT:POP A:DEC A:JR NZ,LOOP:PUSH A
170 "ENDE POP A:RCF:RET
180 "INIT OR. (&D00D),&7F:AND. (&D00F),0
190 " OR. (&D00B),&C0:AND. (&D00A),&FD
200 " RET
210 "COUT BIT. (&D00F),&80:JR Z,COUT
220 " XOR A,&FF:SRA:LD. (&D00F),A
230 " IF#C:OR. (&D00B),&20:ELSE#:AND. (&D00B),&DF:ENDIF
240 " OR. (&D00B),&40:AND. (&D00B),&BF
250 " RET
260 "END

```

CE-130T

SERVICE MANUAL

CODE : 00ZCE130TS/ME



MODEL CE-130T

RS-232C Level converter

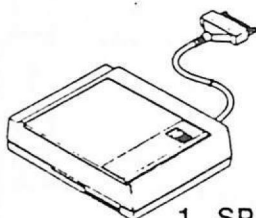
SERVICE MANUAL

SHARP

CODE : 00ZCE140F/SME

MODEL CE-140F

POCKET DISK DRIVE



1. SPECIFICATIONS

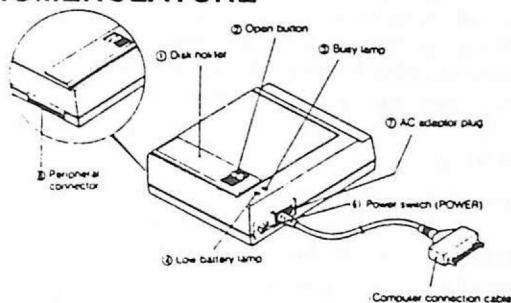
Model name:	CE-140F
Product name:	Pocket disk drive
Number of drives:	Single drive (single sided)/unit
Recording medium:	2.5-inch double-sided disk
Recording method:	GCR (4/5)
Number of tracks:	16 tracks/side (8 sectors/track, 512 bytes/sector)
Capacity:	64K bytes (single side) 128K bytes (double sides) The user area size (at formatting) is 62464 Bytes (Single side).
Power source:	7.5V ... (DC) Dry cell, size AA (or R6) x 5 AC: AC adaptor (EA-160)
Power consumption:	2.5W
Cell life:	Approx. 60 minutes for manganic cell SUM-3(C). R6P Approx. 200 minutes for alkaline cell AM-3.LR6 (These values are obtained when a 4K-byte program is written and read continuously at a temperature of 20°C.) The life of the dry cell may be shorter than the above values because of natural discharge. In addition; it may vary slightly depending on environ- ment and use.
Temperature:	10°C to 35°C (environmental condi- tions for drive operation)
Humidity:	20% to 80% (without condensation)

Do not sale !

Dimension:	118(width) × 145 (depth) × 39 (height) mm
Weight:	Approx. 650g (including cells)
Accessory:	Pocket disk (1), dry cell (5), operation manual (1)
Option:	CE-1650F (ten disks) EA-160 (adaptor)

Note: A 2.5-inch (approx. 63.5mm) disk indicates that the diameter of the disk is 2.5 inches.

2. NOMENCLATURE



3. NOTE ON USE

Because the pocket disk drive is a precision instrument, use and store it observing the following points. Ignoring these precautions will result in the pocket disk drive being damaged.

- ① Do not turn the power on or off when a disk is in the pocket disk drive.
- ② Do not push the open button while the busy lamp is lit. Pushing this button may cause data to be deleted from the disk.
- ③ If the low battery lamp (red) lights during disk operation, turn off the power after completion of the operation, replace the dry cells, or connect the AC adaptor.
- ④ Connect the pocket disk drive as specified and use it in a safe, vibration-free place.
- ⑤ Connect or disconnect all devices, including the pocket disk drive, only after setting their power switches to OFF.
- ⑥ Do not jolt or put weight on the pocket disk drive. Especially, avoid putting heavy objects on the top cover of the disk drive.
- ⑦ Remove the disk and dry cells when the pocket disk drive is not to be used for a long time.

4. NOTES

- Data compatibility

Data may not be read because of data incompatibility between the PC-1360 and PC-1460, PC-1403 or PC-1425, as shown below.

1. ASCII files (programs saved by SAVE "file name", A) can be used by all machines.
 2. Data files can be used by all machines.
 3. Intermediate files (binary files saved by SAVE "file name") can be used by both the PC-1460, PC-1403 and PC-1425, but not by the PC-1360 (ERROR 9).
- When two disks are being used for COPY command execution, if the **BRK** key is pressed to interrupt the execution, turn off the power of the pocket disk drive, and turn it on again after five seconds.
 - Check whether the disk has been write-inhibited before using it.

5. INSTRUCTIONS FOR CE-140F

CHAIN	INIT	MERGE
CLOSE	KILL	NAME
COPY	LFILES	OPEN
DSKF	LOAD	PRINT#
EOF	SAVE	SET
FILES	LOC	
INPUT#	LOF	

CENTRONICS selbstgemacht

für den Sharp 1401/1402

Das UVO ist eine feine Sache, nur leider sehr teuer und unendlich langsam, denn es muß ja die gesamte Arbeit des Dekodierens übernehmen. Was liegt näher, als dem Sharp selbst diese Aufgabe zu übertragen. Ein bißchen pfiffig eingesetzte Maschinensprache und ein Minimum an Hardware lassen im Nu ein völlig normgerechte Centronicsinterface entstehen, das erstens recht billig und zweitens sehr schnell ist. Wenn der Drucker über einen genügend großen Puffer verfügt, ist auch ein großes Programm in weniger als fünf Sekunden übertragen und der Sharp oder der Programmierer kann sich anderen Aufgaben widmen. Mit dem Interface hat man endlich Listings mit 80 Zeichen pro Zeile, die man auch lesen kann, ohne sich die Augen zu zermartern, weil die Zeile nach 24 Buchstaben umgebrochen wird.

Das Prinzip: Bit für Bit

Das Prinzip ist denkbar einfach. Zum Einsatz kommt ein 8-Bit-Schieberegister mit serielllem Eingang und parallelem Ausgang, das die Seriell/Paralell-Wandlung übernimmt. Angesteuert wird der Baustein über die Sharp-Schnittstelle. Dazu legt der Sharp ein Datenbit an den Dateneingang des 74HCT164 und taktet danach den Clock-Eingang. Bei jedem Takt wird das Bit um eine Stelle verschoben. Sind acht Bit getaktet, wird der Busy-Ausgang des Druckers solange abgefragt, bis der Drucker Bereitschaft signalisiert. Dann wird über den Inverter 74HC14 das Strobe-Signal erzeugt, mit dem dem Drucker gesagt wird, daß an den parallelen Dateneingängen ein gültiges Zeichen anliegt.

Der Rest: Klartext

Ein Byte über ein normgerechtes Interface zu schicken, ist schon eine feine Sache. Was man aber wirklich will, ist sich ein komplettes Listing ausdrucken zu lassen oder eine Printanweisung auf den Drucker anzuwenden. Die BASIC-Befehle LPRINT und LLIST helfen da nicht weiter, weil sie auf SHARP-Geräte zugeschnitten sind. Auch hier verhilft die Maschinensprache zu sehr schnellen Lösungen. Das hier vorgestellte Programm liest den Programmtext und konvertiert automatisch das geräteinterne Format in lesbaren Klartext. Auch Printanweisungen können auf die Schnittstelle umgelenkt werden.

Wer sich für die Einzelheiten der Programmtechnik interessiert, lese sich die Kommentare im Assemblerlisting gut durch. Mit Sicherheit wird er Anregungen finden für eigene Programme. Dem nicht Maschinensprache-Fan sei gesagt: es funktioniert tatsächlich.

Die Software

Was ist zu tun, wenn man selbst in den Genuß eines Centronics-interfaces kommen will. Zuerst die Software.

Da das Maschinenprogramm ständig im Speicher sein soll und nicht mit dem Basicspeicher kollidieren soll, müssen Sie den Basicanfang verschieben, bevor Sie das Installationsprogramm eingetippen. So bleibt das Programm ständig im Speicher und übersteht auch einen NEW-Befehl. Wer eine Speichererweiterung für Maschinensprache besitzt, packt das Programm natürlich dorthin. Verschoben wird der Basicanfang im Pro-Modus mit

POKE &46E1,0,&21<ENTER> für den Sharp PC-1402
POKE &46E1,0,&39<ENTER> für den Sharp PC-1401

und NEW<ENTER>

Jetzt kann das Installationsprogramm eingetippt werden und, nachdem es gründlich kontrolliert wurde, mit RUN gestartet werden. Es fragt nach der Anfangsadresse des Bereiches, in das der Druckertreiber geschrieben werden soll und gibt nach Beendigung des Programms die Adressen, die anstatt der Befehle LLIST und LPRINT geCALLet werden müssen. Außerdem bestimmt es eine Checksumme, um Fehleingaben zu erkennen. Obwohl das Treiberprogramm nicht verschiebbar ist, paßt das Installationsprogramm die entsprechenden Adressen an. Spätestens nach dem ersten Programmdurchlauf sollte eine Sicherheitskopie sowohl des Installationsprogramms mit CSAVE bzw. des Treibes mit CSAVE M Adr,Adr+&D0 gemacht werden.

Wenn die Hardware soweit ist, wird's Ernst. Ein Listing erhalten Sie mit dem Maschinenprogrammaufruf:

CALL Adr

POCKET COMPUTER

Die Printanweisung sieht so aus:

```
WAIT 0:PRINT Argumentenliste:CALL Adr
```

wobei für Adr der jeweilige Wert einzusetzen ist, der vom Installationsprogramm angezeigt wurde. Ich habe mir die Adressen auf die Innenseite des Schubers geklebt, so daß ich sie jederzeit parat habe. Die Argumentenliste hinter dem PRINT-Statement kann beliebig sein. Solange das Sharpbasic nicht meckert, verkraftet es auch das Interface. Allerdings werden TOKEN, die in Stringvariablen stehen, nicht detokenisiert.

Ein Problem: Steuerzeichen

Um einen Drucker richtig nutzen zu können, möchte man natürlich auch softwaregesteuert z.B. zwischen verschiedenen Zeichensätzen umschalten. Leider verweigert das Sharp-BASIC hier die Mitarbeit, weil hierzu meist ESCAPE-Sequenzen benutzt werden und das BASIC bei PRINT CHR\$(27) meckert. Aber vielleicht unterscheidet der Drucker nicht zwischen einem normalen ESCAPE und einem ESCAPE mit gesetztem achten Bit. Das steht nämlich auf der Tastatur als DEG zur Verfügung. Hier hilft nur ein Studium des Druckerhandbuches und Probieren.

Die Hardware: klein aber fein

Der Schaltplan erläutert, wie die einzelnen Bauelemente verschaltet werden. Im Prinzip reicht eine kleine Lochrasterplatine, um die einzelnen Bauteile aufzunehmen. Verdrahtet werden sie dann mit isoliertem Lackdraht. Als Gehäuse eignet sich sehr gut ein C64-Userportgehäuse, das dem Interface ein professionelles Aussehen gibt. Wer sich die Arbeit des Selbstverdrahtens nicht zutraut, kann über die Fischel GmbH einen kompletten Bausatz bestellen, der allen nötigen Teile enthält, nämlich:

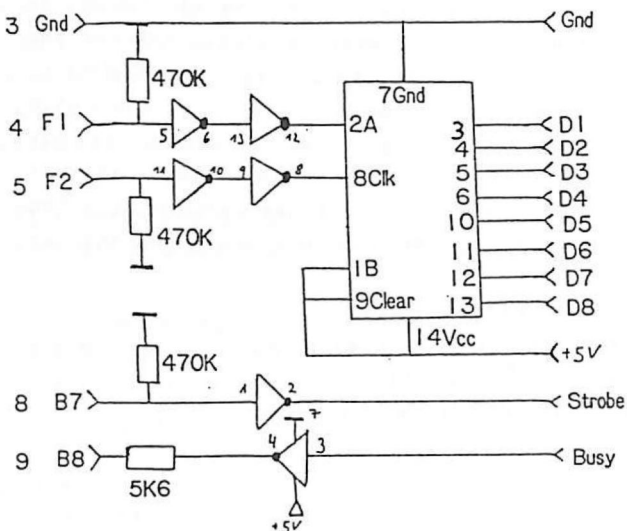
- eine geätzte und gebohrte Platine
- ein Schieberegister 74HC164
- ein 6-fach Inverter 74HCT14
- drei Pulldownwiderstände 470K
- ein Schutzwiderstand 5K6
- ein 11-poligen gewinkelten Pfostenstecker vergoldet
- ein Userportgehäuse
- 12-adriges Flachkabel
- ein Centronicsstecker
- Lötzinn

Do not sale

Die Stromversorgung

Die verwendete CMOS-Technologie beschränkt den Strombedarf auf ein Minimum. Es ist deshalb möglich, das Interface vom Drucker aus mit Strom zu versorgen. Die meisten Drucker (z.B. NEC P6, Star SG10) stellen dazu 5 Volt an Pin 18 zur Verfügung. Schauen Sie ins Druckerhandbuch, ob Pin 18 tatsächlich 5 Volt führt und auch nicht anderweitig belegt ist. Ansonsten müßte die Spannungsversorgung mit Kabeln extern zugeführt werden.

Obwohl die Schaltung nach bestem Wissen an zwei verschiedenen Druckern ausprobiert wurde, haften ich nicht für eventuelle Schäden, die durch die Benutzung des Interfaces entstehen.



Das Installationsprogramm

```

1 "CENTRONICS-TREIBER FUER DEN SHARP PC 1401/1402 BY BO-FU
10 DATA &84,&10,&46,&E1,&1A,&24,&67,&FF,&38,&A0,&6B,&08,&28,&9C,&34,&00
20 DATA &07,&23,&89,&1E,&5B,&DA,&C3,&3A,&0B,&02,&56,&90,&0C,&02,&02,&8F
30 DATA &0C,&2D,&0C,&24,&DA,&C3,&3A,&07,&02,&01,&90,&0C,&2D,&08,&88,&02
40 DATA &05,&DB,&C8,&23,&8E,&DB,&74,&30,&C9,&28,&06,&67,&30,&38,&05,&C8
50 DATA &78,&AH,&AL,&00,&02,&90,&1D,&49,&29,&17,&04,&02,&20,&34,&78,&AH
60 DATA &AL,&5B,&67,&0D,&28,&06,&78,&BH,&BL,&2D,&55,&24,&5A,&3A,&04,&D2
70 DATA &2D,&14,&67,&F6,&28,&03,&02,&5C,&67,&F8,&28,&03,&02,&28,&84,&74
80 DATA &C6,&DB,&34,&23,&5A,&74,&A9,&85,&DB,&34,&24,&04,&85,&DB,&84,&55
90 DATA &05,&24,&67,&7F,&2A,&06,&78,&AH,&AL,&2D,&09,&85,&5B,&DB,&84,&5B
100 DATA &DB,&2D,&47,&02,&AF,&84,&DB,&85,&02,&47,&DB,&24,&34,&78,&AH,&AL
110 DATA &5B,&67,&0D,&29,&09,&02,&0A,&2C,&03,&02,&0C,&00,&08,&12,&5E,&60
120 DATA &FC,&5A,&2A,&03,&61,&02,&5F,&61,&01,&5F,&41,&29,&0D,&60,&FC,&5F
130 DATA &6B,&08,&28,&05,&CC,&5A,&2B,&07,&12,&5D,&61,&40,&DD,&60,&BF,&DD
140 DATA &37
150 S=0:IA=&3800:INPUT "ANF-ARESSE:":IA
160 A=IA+&AB:B=IA+&A5:AH=INT (A/256):AL=A-AH*256:BH=INT (B/256):BL=B-BH*256
170 SU=&3E56+4*(AH+AL)+BH+BL
180 FOR I=IA TO IA+&DO:READ X:POKE I,X:S=S+X:NEXT I
190 IF S<>SU THEN PRINT "DATA-ERROR":END
200 PAUSE "DRUCKER-TREIBER":PAUSE "INSTALLIERT"
210 PRINT " LLIST=CALL&":HEX (IA)
220 PRINT " LPRINT=CALL&":HEX (IA+&93)
    
```

POCKET COMPUTER

Centronicstreiber für den Sharp PC 1402 und 1401
 von Ansgar Boczek-Funcke
 Schauenburger Str.54
 2300 Kiel 1

```

3800: 84      LPO4      ; Highbyte des Basicanfags
3801: 10 46 E1 LIDP 46E1 ; aus 46E1,46E2
3804: 1A      MVBD      ; nach Xh laden
3805: 24      IXL       ; erstes Byte lesen
3806: 67 FF   CPIA FF   ; auf Programmende testen
3808: 38 A0   JRZP -38A9 ; Sprung wenn Programmende erreicht
380A: 6B 8    TEST 8    ; Break gedrückt?
380C: 28 9C   JRNZP -38A9; Sprung, wenn ja
380E: 34      PUSH     ; erste Byte auf Stack retten
380F: 0 7     LII 7     ; I-Register Zähler für 7 Byte
3811: 23      CLRA     ; Akku löschen
3812: 89      LPO9     ; internes RAM von L-Register bis 10
3813: 1E      FILM    ; Null setzen ; OE OF 10=BCD Buffer
3814: 5B      POP      ; Byte vom Stack holen
3815: DA      EXAB    ; ins B-Register
3816: C3      DECB    ; B=B-1
3817: 3A B    JRCP -3823 ; Sprung wenn Schleife zu Ende (B=6FF)
3819: 2 56    LIA 56    ; Akku mit 56 laden
381B: 90      LP10    ; B mal BCD addition von 56
381C: C       ADN     ; zum BCD Buffer
381D: 2 2     LIA 2     ; Akku mit zwei laden
381F: 8F      LPOF    ; Zeiger auf drittletzte Stelle
3820: C       ADN     ; BCD Addition=Addition von 200
3821: 2D C    JRM -3816 ; Schleifensprung
3823: 24      IXL     ; lowByte der Zeilennummer lesen
3824: DA      EXAB    ; Akkuinhalt ins K-register
3825: C3      DECB    ; B=B-1
3826: 3A 7    JRCP -382E ; Sprung wenn Schleife zu Ende
3828: 2 1     LIA 1     ; Akku=1 setzen
382A: 90      LP10    ; P zeigt auf letzte Stelle des Buffers
382B: C       ADN     ; BCD-Addition (B mal)
382C: 2D 8    JRM -3825 ; Schleifensprung
382E: 88      LPO8    ; K=5; Zähler für max 5 BCD Stellen
382F: 2 5     LIA 5     ;
3831: DB      EXAM    ;
3832: C8      INCL    ; L=Vornullenflag setzen
3833: 23      CLRA     ; Akku löschen
3834: 8E      LPOE    ; erste Stelle des BCD-Buffers
3835: DB      EXAM    ; in Akku laden
3836: 74 30   ADIA 30  ; Asciioffset addieren
3838: C9      DECL    ; L=L-1; wird 0 wenn Vornull möglich
3839: 28 6    JRNZP -3840; Sprung wenn keine Vornull
383B: 67 30   CPIA 30  ; mit "0" vergleichen
383D: 38 5    JRZP -3843 ; Sprung wenn ja
383F: C8      INCL    ; ansonsten Vornullenflag überhoch setzen
3840: 78 38 AB CALL -38AB ; CALL "Out"
3843: 0 2     LII 2     ; den 3 Byte langen
3845: 90      LP10    ; BCD-Buffer
3846: 1D      SLW     ; um ein Nipple nach links schieben
3847: 49      DECK    ; K=K-1
3848: 29 17   JRNZM -3832; Schleifensprung bis letzte Stellen ausgegeben
384A: 4       IX      ; Zeilenlängenbyte übergehen
384B: 2 20    LIA 20    ; Blank laden
384D: 34      PUSH    ; auf Stack retten
384E: 78 38 AB CALL -38AB ; CALL "Out"
3851: 5B      POP      ; vom Stack holen
3852: 67 D    CPIA D    ; auf Zeilenende testen
3854: 28 6    JRNZP -385B; Rücksprung wenn nicht Zeilenende
3856: 78 38 A5 CALL -38A5 ; CALL "Linefeed"
3859: 2D 55   JRM -3805 ; Rücksprung nächste Zeile
385B: 24      IXL     ; nächstes Byte lesen
385C: 5A      SL      ; höchstes Bit in Carry schieben
385D: 3A 4    JRCP -3862 ; Sprung wenn Token, Sonderzeichen
385F: D2      SR      ; ansonsten Byte wiederherstellen
3860: 2D 14   JRM -384D ; Rücksprung zum Byte auf Stack legen
3862: 67 F6   CPIA F6   ; Test Sonderzeichen Pi
3864: 28 3    JRNZP -3868; Sprung wenn nicht Pi
3866: 2 5C    LIA 5C    ; sonst Token für Klartext PI laden
3868: 67 F8   CPIA F8   ; Test Sonderzeichen Wurzel
386A: 28 3    JRNZP -386E; Sprung wenn kein Wurzel
386C: 2 28    LIA 28    ; sonst Token für Klartext SQR laden
386E: 84      LPO4     ;
386F: 74 C6(62) ADIA C6 ; &C6=lowByteoffset zu Token addieren (62=alter 1401)
3871: DB      EXAM    ; alte Adresse im Listing mit Token-
3872: 34      PUSH    ; tabelle tauschen und auf Stack retten
3873: 23      CLRA     ; Akku löschen
3874: 5A      SL      ; eventuellen Obertrag aus 1. Addition zum Akku
3875: 74 A9   ADIA A9 ; &A9= highbyteOffset der Tokentabelle addieren
3877: 85      LPO5     ; mit alter Adresse
3878: DB      EXAM    ; tauschen
3879: 34      PUSH    ; und auf Stack retten
387A: 24      IXL     ; Highbyte des Anfangsadresse Tokenstring zum Akku
387B: 4       IX      ; auf Lowbyte zeigt DP
387C: 85      LPO5     ; Highbyte nach
387D: DB      EXAM    ; Xh
    
```

Do not sale

387E: 84	LPO4	; Lowbyte
387F: 55	MVMD	; nach X1
3880: 5	DX	; X ein Byte vor Tokenstring positionieren
3881: 24	IXL	; Byte lesen
3882: 67 7F	CPIA 7F	; Stringende erreicht?
3884: 2A 6	JRNCP -388B	; Sprung wenn ja
3886: 78 38 AB	CALL 38AB	; CALL "Out"
3889: 2D 9	JRM -3881	; Rücksprung nächstes Byte lesen
388B: 85	LPO5	; Highbyte der alte Adresse
388C: 5B	POP	; vom Stack
388D: DB	EXAM	; nach Xh
388E: 84	LPO4	; Lowbyte der alten Adresse
388F: 5B	POP	; vom Stack
3890: DB	EXAM	; nach X1
3891: 2D 47	JRM -384B	; Rücksprung nächste Byte im Listing
3893: 2 AF	LIA AF	; AF= lowbyte des Ausgabepuffers
3895: 84	LPO4	; nach X1
3896: DB	EXAM	; laden
3897: 85	LPO5	; nach Xh
3898: 2 47	LIA 47	; die 47
389A: DB	EXAM	; laden ; X=447AF=Anfangsadresse-1 des Printbuffers
389B: 24	IXL	; Byte lesen
389C: 34	PUSH	; auf Stack retten
389D: 78 38 AB	CALL -38AB	; CALL "Out"
38A0: 5B	POP	; vom Stack holen
38A1: 67 D	CPIA D	; auf Zeilenende testen
38A3: 29 9	JRNZM -389B	; Sprung wenn kein Zeilenende
38A5: 2 A	LIA A	; Akku mit Linefeed laden
38A7: 2C 3	JRP.-38AB	; und ausgeben, gleichzeitig RETURN zum Basic
38A9: 2 C	LIA C	; Akku mit FORMFEED laden
38AB: 0 8	LII 8	; I=Zähler für 8 Bit
38AD: 12 5E	LIP 5E	; im F-Port
38AF: 60 FC	ANIM FC	; Bit 1 und 2 löschen
38B1: 5A	SL	; Datenbit ins Carry schieben
38B2: 2A 3	JRNCP -38B6	; Sprung wenn low
38B4: 61 2	ORIM 2	; Bit 2 setzen
38B6: 5F	OUTF	; und nach Pin 5 ausgeben (DATA des Schieberegisters)
38B7: 61 1	ORIM 1	; Bit 1 setzen (Takt für Schieberegister)
38B9: 5F	OUTF	; und ausgeben
38BA: 41	DECI	; Bitzähler erniedrigen
38BB: 29 D	JRNZM -38AF	; Schleifenpunkt bis 8 Bit getaktet
38BD: 60 FC	ANIM FC	; Bit 1+2 in F löschen
38BF: 5F	OUTF	; und ausgeben
38C0: 6B 8	TEST 8	; Break gedrückt?
38C2: 28 5	JRNZP -38C8	; Sprung wenn ja
38C4: CC	INB	; B abfragen
38C5: 5A	SL	; Bit 8 ins Carry schieben
38C6: 2B 7	JRNCH -38C0	; Sprung wenn Drucker busy
38C8: 12 5D	LIP 5D	; Bit 7 in B
38CA: 61 40	ORIM 40	; setzen
38CC: DD	OUTB	; und ausgeben
38CD: 60 BF	ANIM BF	; Bit 7 in B löschen
38CF: DD	OUTB	; und ausgeben (STROBE)
38D0: 37	RTN	; Return

Kassette 125.- DM (incl. 14% Mwst.)

RS232C am Sharp 1401/02

Die RS232-Schnittstelle ist die weitverbreitetste Schnittstelle, die von Computer zur Kommunikation untereinander genutzt wird. Fast jeder Computer verfügt heute über eine. Leider nicht so eine Reihe von Sharp-Taschencomputer. Sind sie vielleicht gar keine richtigen Computer, sondern nur aufgemotzte Taschenrechner. Mitnichten. Mit ein bißchen Maschinensprache und einigen Kenntnissen über die Eigenarten der Sharpschnittstelle ist eine softwaremäßige RS232-Schnittstelle für den Sharp kein Problem. Ein wenig Hardware sorgt dann für die nötige Pegelanpassung. Doch damit nicht genug. Da der Sharp sein Programm tokenisiert, d.h. ein reserviertes Schlüsselwort durch ein einziges Byte speichert, ist auch bei normgerechter Übertragung nicht unbedingt gewährleistet, daß der Empfangsrechner (Host) das Programm auch ohne weiteres verarbeiten kann.

Auch hier hilft die Maschinensprache. Das hier vorgestellte Programm kann noch viel mehr. Komplett in Maschinensprache geschrieben unterstützt es folgenden Punkte:

- Senden eines Basicprogramms im Klartext
- Umlenken der Printanweisung auf die serielle Schnittstelle
- Einlesen von an der Schnittstelle empfangenen Daten in eine (Feld-)variable
- Und zur Krönung: Empfangen eines Programmes im Klartext mit Umwandlung in das interne Format.

Bei soviel Leistungsfähigkeit fragt sich der biedere Benutzer. Muß ich jetzt auch Maschinensprache können? Beileibe nicht. Alle Funktionen sind auch vom Basic aus aufrufbar. Bevor es ans Abtippen geht, muß der Basicanfang im Pro-Modus mit

POKE &46E1,0,&22 <ENTER> für den 1402

bzw. POKE&46E1,0,&40 für den 1401

und

NEW <ENTER> 1401/1402

verschoben werden. So bleibt das Programm ständig im Speicher und übersteht auch einen NEW -Befehl.

Jetzt kann das entsprechende Installationsprogramm abgetippt werden und nach sorgfältiger Kontrolle mit RUN gestartet werden.

~~Do not sell~~

~~POCKET COMPUTER~~

Das Installationsprogramm berechnet eine Checksumme, um Fehleingaben zu vermeiden. Und gibt danach die Adresse aus, die geCALLet werden müssen, um die verschiedenen Funktionen zu erfüllen. Im folgenden bezieht sich Adr auf diese Werte, die man sich am besten gleich notiert und an die Innenseite des Schubers klebt. So hat man sie immer zur Hand. Die Übertragung läßt sich mit BREAK unterbrechen. Wartet der Sharp auf ein Startbit, muß die Übertragung dadurch unterbrochen werden, daß das Interface kurz stromlos gemacht wird, da aufgrund der sonst fehlenden Synchronisation hier keine Tastaturabfrage möglich war. Andere Baudraten können durch Ändern des WAIT Arguments (siehe Assembler-Listing) eingestellt werden.

Ein komplettes Programm wird mit CALL Adr(LIST#) übertragen.

Die Printanweisung auf die Schnittstelle sieht so aus

WAIT 0:PRINT Argumentenliste:CALL Adr(PRINT#)

Erwartet der Host-Rechner zum Abschluß der Übertragung ein CTRL-Z, kann dieses mit CALL Adr(CTRL-Z) gesendet werden.

Zum Einlesen von Variablen muß eine Stringvariable als Feldvariablenpuffer mit der zu erwartenden Länge vereinbart werden und direkt vor dem Einlesen initialisiert werden.

Also z.B.:CLEAR:DIM PUS(0)*80:PUS(0)="" : CALL Adr(INPUT#)

Danach steht in der Variablen PUS(0) eine komplette Zeile, die danach beliebig weiterverarbeitet werden kann.

Ein komplettes Programm im Klartext empfangen kann man mit einem simplen CALL Adr(MERGE#). Das empfangene Programm wird zeilenweise eingelesen und danach tokenisiert. Die Zeilen können bis zu 157 Zeichen lang sein. Allerdings wird nicht geprüft, ob sich die Zeichenanzahl durch die Tokensierung auf die maximal zulässigen 80 Zeichen reduziert. Am besten werden nur maximal 80 Zeichen in einer Zeile gesendet. Das Programm wird an ein bereits im Speicher befindliches angehängt. Allerdings wird nicht überprüft, ob die Zeilennummern in aufsteigender Reihenfolge erscheinen. Also Vorsicht. Wenn am Beginn der Zeile keine Zeilennummer stehen, gilt die Übertragung als beendet.

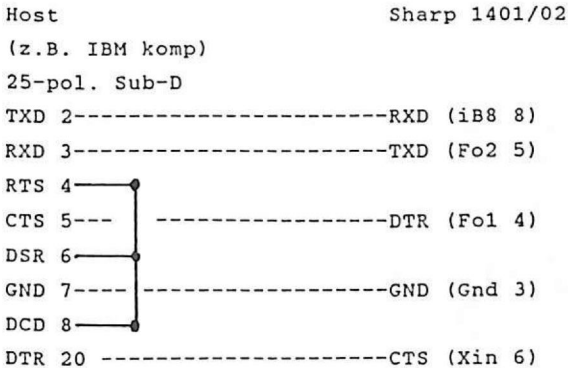
Auf dem Hostrechner muß lediglich ein Programm vorhanden sein, daß an der seriellen Schnittstelle mit 4800 BAUD,8Bit,2 Stopbits, keine Parität senden und empfangen kann. Oft tut es auch das Betriebssystem oder ein kleine Basicprogramm. Einige Rechner senden nach Erkennen des Busysignals noch weitere Zeichen (IBM-

kompatible zum Beispiel noch zwei bei 4800 BAUD). Der Sharp nimmt als Zeilenendmarkierung allerdings nur das CR (&OD) an. Das Sendeprogramm sollte sich an diese Konvention halten und am Zeilenende ein CR(&OD) schicken und dann sofort den HANDSHAKE-Status lesen, damit bei der Übertragung keine Zeichen verloren gehen.

Die Hardware

Die Hardware kann sich auf die Pegelwandlung beschränken. Die beschriebene Schaltung tut diese Aufgabe klaglos. Zum Aufbau reicht eine Lochrasterplatine, die mit Lackdraht verkabelt wird. Wer geschickt ist, bekommt alles in ein C64-Userportgehäuse, so daß der Pegelwandler nachher richtig professionell aussieht. Und nun viel Spaß beim Abtippen und Löten. Willkommen in der RS232-Welt.

Die richtige Verbindung:



Das Programm und die Schaltung wurden sorgfältig getestet und an verschiedenen Rechnern ausprobiert. Allerdings kann ich bei eventuellen Fehlern oder Hardwareschäden keine Haftung übernehmen. Das Programm ist für Sharp-User geschrieben. Es darf ohne schriftliche Genehmigung des Autor nicht zu kommerziellen Zwecken verbreitet oder vertrieben werden.

- | | |
|-----------------------|----------------------|
| R1,2 470K | C5,6 100nF ker. |
| R3,4 5K6 | IC1 ICL232 o. MAX232 |
| C1..4 22uF Tantal | IC2 78L05 |
| o.1uF Tantal (ICL232) | IC3 40106 |
| | D1 1N4148 |

POCKET COMPUTER

sonst.: Platine

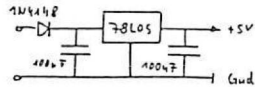
11 polige Steckerleiste gewinkelt

C64-Userportgehäuse

Kabel 5-polig

25-pol-Sub-D Stecker

Klinkenbuchse 2.5 mm



Sharp PC1401/02

Pin

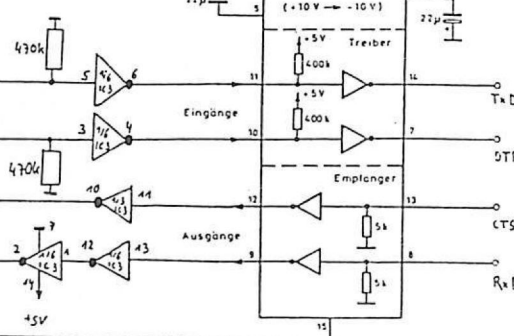
F02 5

F01 4

Xin 6

IBS 8

GND 3



```

:
: RS232C Treiber für den Sharp PC 1402
: von Ansgar Boczek-Funcke
: Schauenburger Str.54
: 2300 Kiel 1
:

```

2000:84	LP04	; Highbyte des Basicanfags
2001:10 46 E1	LIDP 46E1	; aus 46E2
2004:1A	MVBD	; nach Xh laden
2005:24	IXL	; erstes Byte lesen
2006:67 FF	CPIA FF	; auf Programmende testen
2008:38 A0	JRZP S20A9	; Sprung wenn Programmende erreicht
200A:6B 8	TEST 8	; Break gedrückt?
200C:28 9C	JRNZP S20A9	; Sprung, wenn ja
200E:34	PUSH	; erste Byte auf Stack retten
200F:0 7	LII 7	; I-Register Zähler für 7 Byte
2011:23	CLRA	; Akku löschen
2012:89	LPOS	; internes RAM von L-Register bis 10
2013:1E	FILM	; Null setzen ; OE OF 10=BCD Buffer
2014:5B	POP	; Byte vom Stack holen
2015:DA	EXAB	; ins B-Register
2016:C3	DECB	; B=B-1
2017:3A B	JRCP S2023	; Sprung wenn Schleife zu Ende (B=&FF)
2019:2 56	LIA 56	; Akku mit 56 laden
201B:90	LP10	; B mal BCD addition von 56
201C:C	ADN	; zum BCD Buffer
201D:2 2	LIA 2	; Akku mit zwei laden
201F:8F	LPOF	; Zeiger auf drittletzte Stelle
2020:C	ADN	; BCD Addition=Addition von 200
2021:2D C	JRM 62016	; Schleifensprung
2023:24	IXL	; lowByte der Zeilennummer lesen
2024:DA	EXAB	; Akkuinhalt ins K-register
2025:C3	DECB	; B=B-1
2026:3A 7	JRCP S202E	; Sprung wenn Schleife zu Ende
2028:2 1	LIA 1	; Akku=1 setzen
202A:90	LP10	; P zeigt auf letzte Stelle des Buffers
202B:C	ADN	; BCD-Addition (B mal)
202C:2D 8	JRM S2025	; Schleifensprung
202E:88	LPO8	; K=5; Zähler für max 5 BCD Stellen
202F:2 5	LIA 5	; ;
2031:DB	EXAM	; ;
2032:C8	INCL	; L=Vornullflag setzen
2033:23	CLRA	; Akku löschen
2034:8E	LPOE	; erste Stelle des BCD-Buffers
2035:DB	EXAM	; in Akku laden
2036:74 30	ADIA 30	; Asciioffset addieren
2038:C9	DECL	; L=L-1; wird 0 wenn Vornull möglich
2039:28 6	JRNZP S2040	; Sprung wenn keine Vornull
203B:67 30	CPIA 30	; mit "0" vergleichen

PCSKET COMPUTER

```

203D:38 5   JRZP $2043 ; Sprung wenn ja
203F:C8     INCL      ; ansonsten Vornullenflag überhoch setzen
2040:78 20 AB CALL $20AB ; CALL "Out"
2043:0 2    LII 2     ; den 3 Byte langen
2045:90     LPI0     ; BCD-Buffer
2046:1D     SLW      ; um ein Nipple nach links schieben
2047:49     DECK     ; K=K-1
2048:29 17  JRNZM $2032; Schleifensprung bis letzte Stellen ausgegeben
204A:4      IX       ; Zeilenlängenbyte übergehen
204B:2 20   LIA 20    ; Blank laden
204D:34     PUSH     ; auf Stack retten
204E:78 20 AB CALL $20AB ; CALL "Out"
2051:5B     POP      ; vom Stack holen
2052:67 D   CPIA D    ; auf Zeilenende testen
2054:28 6   JRNZP $205B; Rücksprung wenn nicht Zeilenende
2056:78 20 A5 CALL $20A5 ; CALL "Linefeed"
2059:2D 55  JRM $2005 ; Rücksprung nächste Zeile
205B:24     IXL     ; nächstes Byte lesen
205C:5A     SL      ; höchstes Bit in Carry schieben
205D:3A 4   JRCP $2062 ; Sprung wenn Token,Sonderzeichen
205F:D2     SR      ; ansonsten Byte wiederherstellen
2060:2D 14  JRM $204D ; Rücksprung zum Byte auf Stack legen
2062:67 F6  CPIA F6   ; Test Sonderzeichen Pi
2064:28 3   JRNZP $2068; Sprung wenn nicht Pi
2066:2 5C   LIA 5C    ; sonst Token für Klartext PI laden
2068:67 F8  CPIA F8   ; Test Sonderzeichen Wurzel
206A:28 3   JRNZP $206E; Sprung wenn kein Wurzel
206C:2 28   LIA 28    ; sonst Token für Klartext SQR laden
206E:84     LPO4     ;
206F:74 C6  ADIA C6   ; &C6=lowByteoffset zu Token addieren
2071:DB     EXAM     ; alte Adresse im Listing mit Token-
2072:34     PUSH     ; tabelle tauschen und auf Stack retten
2073:23     CLRA    ; Akku löschen
2074:5A     SL      ; eventuellen Übertrag aus 1. Addition zum Akku
2075:74 A9  ADIA A9   ; &A9= highByteOffset der Tokentabelle addieren
2077:85     LPO5     ; mit alter Adresse
2078:DB     EXAM     ; tauschen
2079:34     PUSH     ; und auf Stack retten
207A:24     FXL     ; Highbyte des Anfangsadresse Tokenstring zum Akku
207B:4      IX       ; auf Lowbyte zeigt DP
207C:85     LPO5     ; Highbyte nach
207D:DB     EXAM     ; Xh
207E:84     LPO4     ; Lowbyte
207F:55     MVMD    ; nach Xl
2080:5      DX      ; X ein Byte vor Tokenstring positionieren
2081:24     IXL     ; Byte lesen
2082:67 7F  CPIA 7F   ; Stringende erreicht?
2084:2A 6   JRNCP $208B; Sprung wenn ja
2086:78 20 AB CALL $20AB ; CALL "Out"
2089:2D 9   JRM $2081 ; Rücksprung nächstes Byte lesen
208B:85     LPO5     ; Highbyte der alte Adresse
208C:5B     POP      ; vom Stack
208D:DB     EXAM     ; nach Xh
208E:84     LPO4     ; Lowbyte der alten Adresse
208F:5B     POP      ; vom Stack
2090:DB     EXAM     ; nach Xl
2091:2D 47  JRM $204B ; Rücksprung nächste Byte im Listing
2093:2 AF   LIA AF   ; AF= lowbyte des Ausgabepuffers
2095:84     LPO4     ; nach Xl
2096:DB     EXAM     ; laden
2097:85     LPO5     ; nach Xh
2098:2 47   LIA 47   ; die 47
209A:DB     EXAM     ; laden ; X=&47AF=Anfangsadresse-1 des Printbuffers
209B:24     IXL     ; Byte lesen
209C:34     PUSH     ; auf Stack retten
209D:78 20 AB CALL $20AB ; CALL "Out"
20A0:5B     POP      ; vom Stack holen
20A1:67 D   CPIA D    ; auf Zeilenende testen
20A3:29 9   JRNZM $209B; Sprung wenn kein Zeilenende
20A5:2 A    LIA A     ; Akku mit Linefeed laden
20A7:2C 3   JRP $20AB ; und ausgeben, gleichzeitig RETURN zum Basic
20A9:2 1A   LIA 1A    ; Akku mit CTRL-Z laden
20AB:2C 1   JRP $20AD ;
20AD:12 5F  LIP 5F   ; P=Controlwort für Xin
20AF:61 40  ORIM 40   ; Xin aktiv setzen
20B1:DF     OUTC     ;
20B2:6B 88  TEST 88   ; Host busy oder nicht Break gedrückt?
20B4:39 3   JRZM $20B2 ; Endlosschleife bis Host ready oder Break
20B6:0 8    LII 8     ; I= Zähler für 8 Bit
20B8:12 5E  LIP 5E   ;
20BA:61 3   ORIM 3    ; letztes und vorletztes Bit von 5E setzen
20BC:5F     OUTP     ; und nach Pin 4 (Handshake) bzw Pin 5 (TxD) schreiben
20BD:4E C   WAIT C    ; Wartezeit 4800 Baud Startbit
20BF:D2     SR      ; Bit ins Carry schieben
20C0:2A 14  JRNCP $20D5; Sprung wenn Bit low
20C2:60 FD  ANIM FD   ; zweitletztes Bit in 5E löschen
20C4:4E 4   WAIT 4     ; Ausgleich für Zeilverlust durch Sprünge
20C6:5F     OUTP     ; nach TxD schreiben
    
```

20C7:4D	NOPW	:	Wartezyklus für 4800 Baud	
20C8:4D	NOPW	:		
20C9:41	DECI	:	Bitzähler erniedrigen	
20CA:29 C	JRNZM \$20BF:		Sprung wenn nicht letztes Bit gesendet	
20CC:60 FD	ANIM FD	:	Stopbit low setzen	
20CE:4E D	WAIT D	:	Wartezyklus Stopbit	
20D0:5F	OUTF	:	und ausgeben	
20D1:60 FE	ANIM FE	:	Handshakeleitung	
20D3:5F	OUTF	:	rücksetzen	
20D4:37	RTN	:	RETURN "out"	
20D5:61 2	ORIM 2	:	zweitletztes Bit in 5E setzen	
20D7:2D 12	JRM \$20C6	:	und rückspringen	
20D9:10 45 EB	LIDP 46EB	:	46EB,46EC enthält die Adresse der letzten	
20DC:86	LP06	:	bearbeiteten Variablen	
20DD:1A	MVBD	:	diese Adresse nach Y	
20DE:7	DY	:	das erste Byte einer Feldvariablen enthält	
20DF:6	IY	:	das Längenattribut	
20E0:88	LP08	:	dieses ins K-Register, damit nicht mehr gelsene wird	
20E1:55	MVMD	:	als in die Variable paßt (nur Feldvariablen)	
20E2:49	DECK	:	K=K-1, damit Oberlänge später erkannt werden kann	
20E3:12 5E	LIP 5E	:	LSB von 5E	
20E5:61 1	ORIM 1	:	setzen	
20E7:5F	OUTF	:	und ausgeben (Handshake Bereit)	
20E8:0 8	LII 8	:	I=Zähler für 8 Bit	4
20EA:3 0	LIB 0	:	B,den Zwischenspeicher Null setzen	4
20EC:CC	INB	:	Bit an Pin 8 abholen	2
20ED:5A	SL	:	nach links in Carry geschrieben	2
20EE:3B 3	JRCH \$20EC	:	Sprung wenn kein Startbit gesendet wurde	4
20FO:4E 9	WAIT 9	:	Zeitanpassung Startbit	15
20F2:4E E	WAIT E	:	Anpassung Datenbit 4800 Baud	20
20F4:CC	INB	:	Datenbit abholen	2
20F5:5A	SL	:	in Carry schieben	2
20F6:DA	EXAB	:	Zwischenspeicher zurückholen	3
20F7:D2	SR	:	aus dem Carry in Akku schieben	2
20F8:DA	EXAB	:	wieder zwischenspeichern	3
20F9:41	DECI	:	Bitzähler runterzählen	4
20FA:29 9	JRNZM \$20F2:		Sprung wenn nicht letztes Bit	7
20FC:DA	EXAB	:	Zwischenspeicher zum Akku	3
20FD:26	IYS	:	in Variable ablegen	6
20FE:67 D	CPIA D	:	auf Zeilenende testen	4
2100:38 4	JRZP \$2105	:	Sprung wenn Zeilenende erreicht	7/4
2102:49	DECK	:	K=K-1	4
2103:29 1C	JRNZM \$20E8:		Sprung wenn verbleibender Platz noch ausreicht?	7
2105:12 5E	LIP 5E	:	letztes Bit von 5E	
2107:60 FE	ANIM FE	:	rücksetzen	
2109:5F	OUTF	:	und ausgeben (Handshake busy)	
210A:37	RTN	:	RETURN "Input"	
210B:4D	NOPW	:	die letzten noch ungenutzten Bytes	
210C:4D	NOPW	:		
210D:4D	NOPW	:		
210E:2 47	LIA 47	:		
2110:87	LP07	:		
2111:DB	EXAM	:		
2112:2 60	LIA 60	:		
2114:86	LP06	:		
2115:DB	EXAM	:	Y=4760 Beginn des Textbuffers	
2116:2 9E	LIA 9E	:		
2118:88	LP08	:	K=dez 158,Länge von Text- und Printbuffer zusammen	
2119:DB	EXAM	:	eine Zeile hat also maximal 158 Character	
211A:78 20 E2	CALL \$20E2	:	CALL Zeile einlesen	
211D:2A 4	JRNCP \$2122:		C-Flag gesetzt wenn überlange Zeile	
211F:2 D	LIA D	:	dann Carriage-Return	
2121:52	STD	:	an letzte Stelle	
2122:10 46 E3	LIDP 46E3	:	DP mit Pointer auf Basicende laden	
2125:86	LP06	:	dessen Inhalt nach Y1,Yh	
2126:1A	MVBD	:	en bloc, das benutzt J-Register liegt immer auf 1	
2127:7	DY	:	eins zurück	
2128:2 47	LIA 47	:	4760=Beginn des Textbuffers	
212A:85	LP05	:		
212B:DB	EXAM	:	47 nach Xh	
212C:2 60	LIA 60	:		
212E:84	LP04	:		
212F:DB	EXAM	:	60 nach Xl	
2130:0 F	LII F	:		
2132:23	CLRA	:	Akk löschen	
2133:88	LP08	:	internes RAM von 8 (K-Reg) bis 18	
2134:1E	FILM	:	Null setzen	
2135:24	IXL	:	erstes Byte lesen	
2136:75 30	SBTA 30	:	Asciioffset subtrahieren	
2138:3A C4	JRCP \$21FD	:	Sprung wenn keine Zahl	
213A:67 10	CPIA 10	:	>=10?	
213C:2A C0	JRNCP \$21FD:		Sprung wenn keine Zahl	
213E:90	LP10	:	erstes Byte nach 10	
213F:DB	EXAM	:		
2140:24	IXL	:	nächstes Byte lesen	
2141:67 40	CPIA 40	:	eine Zahl?	
2143:2A 1C	JRNCP \$2160:		Sprung wenn nein	

Do not sale!

POCKET COMPUTER

```

2145:67 30  CPIA 30  ;eine Zahl?
2147:3A 18  JRCP $2160 ;Sprung wenn nein
2149:75 30  SBIA 30  ;Asciioffset subtrahieren
214B:34     PUSH     ;auf Stack retten
214C:82     LPO2     ;B,A
214D:13 10  LIQ 10   ;=
214F:A     MVB      ;(11,10)
2150:0 9    LII 9     ;I= Zähler bis dezimal 9
2152:90     LP10    ;Diese Schleife entspricht Multiplikation mit 10
2153:14     ADB      ;(11,10)=(11,10)+(B,A)
2154:3A A9  JRCP $21FE ;Sprung wenn zu große Zahl
2156:41     DECI     ;I=I-1
2157:29 6   JRNZM $2152 ;Schleifensprung bis I=0
2159:3 0    LIB 0     ;B Null setzen
215B:5B     POP      ;Ziffer vom Stack holen
215C:90     LP10    ;
215D:14     ADB      ;und dazuaddieren
215E:2D 1F  JRM $2140 ;Rücksprung nächste Zahl
2160:91     LP11    ;(11) Highbyte Zeilennummer
2161:DB     EXAM     ;zum Akku
2162:26     IYS      ;und wegspeichern
2163:90     LP10    ;(10) Lowbyte Zeilennummer
2164:DB     EXAM     ;zum Akku
2165:26     IYS      ;und wegspeichern
2166:6      IY       ;Platz für Zeilenlänge erst mal übergehen
2167:10 21 ED LIDP 21ED ;DP mit Buffer für Zeilenlänge laden
216A:87     LPO7     ;Yh
216B:53     MVDM    ;dorthin
216C:86     LPO6     ;Yl
216D:11 EE  LIDL EE  ;in nächstes Byte
216F:53     MVDM    ;schreiben
2170:24     IXL      ;Zeichen lesen
2171:67 D    CPIA D    ;auf Zeilenende testen
2173:38 76  JRZP $21EA ;Sprung wenn nicht Zeilenende
2175:67 20  CPIA 20   ;ein Blank?
2177:39 8   JRZM $2170 ;Sprung wenn ja, Blanks übergehen
2179:67 22  CPIA 22   ;ein " "?
217B:28 C   JRNZP $2188 ;Sprung wenn nein
217D:26     IYS      ;also eine Stringkonstante, wegschreiben
217E:48     INCK     ;Zeilenlängenzähler hochzählen
217F:24     IXL      ;nächstes Zeichen lesen
2180:67 D    CPIA D    ;auf Zeilenende testen
2182:38 67  JRZP $21EA ;Sprung wenn Zeilenende
2184:67 22  CPIA 22   ;auf Stringende testen
2186:29 A   JRNZM $217D ;Sprung wenn Ende nicht erreicht
2188:67 41  CPIA 41   ;< ASC("A")
218A:3A 5B  JRCP $21E6 ;Sprung wenn ja
218C:67 5A  CPIA 5A   ;> ASC("Z")
218E:2A 57  JRNCP $21E6 ;Sprung wenn ja
2190:8A     LPOA     ;also ein Buchstabe
2191:DB     EXAM     ;0A=Zwischenspeicher mit Byte laden
2192:13 4   LIQ 4     ;
2194:8C     LPOC     ;0C,0D Speicher für aktuelle Stelle im Text
2195:A     MVB      ;(0C,0D)=Xl,Xh
2196:84     LPO4     ;A912 ist der Offset für die alphabetische
2197:2 12   LIA 12   ;Tabelle
2199:DB     EXAM     ;Xl=12
219A:85     LPO5     ;
219B:2 A9   LIA A9   ;
219D:DB     EXAM     ;Xh=A9
219E:3 0    LIB 0     ;B=Null setzen
21A0:8A     LPOA     ;
21A1:59     LDH      ;Buchstabe aus Zwischenspeicher lesen
21A2:D1     RC       ;Carry löschen
21A3:5A     SL       ;nach links schieben, entspricht mal 2
21A4:84     LPO4     ;
21A5:14     ADB      ;(05,04)=(05,04)+(B,A)
21A6:24     IXL      ;Highbyte erstes Token mit diesem Anfangsbuchstabe
21A7:67 0   CPIA 0     ;kein Token mit diesem Buchstaben?
21A9:38 36  JRZP $21E0 ;Sprung wenn ja
21AB:4      IX       ;X steht auf Lowbyte des Tokens
21AC:85     LPO5     ;Xh
21AD:DB     EXAM     ;=Highbyte
21AE:84     LPO4     ;Xl
21AF:55     MVMD    ;=Lowbyte
21B0:0 FF  LII FF   ;I=Tokenstringlängenzähler,erstmal Null -1
21B2:5      DX       ;X= eins vor den Tokenanfang
21B3:8F     LPOF     ;P= eins vor interen RAMbuffer
21B4:50     INCP    ;P= P+1
21B5:4      IX       ;X= X+1
21B6:55     MVMD    ;(X) nach (P)
21B7:40     INCI     ;Längenzähler erhöhen
21B8:63 80  CRIM 80   ;achstes Bit gesetzt? =Token selber erreicht
21BA:3B 7   JRCH $21B4 ;Schleifensprung wenn nein
21BC:4      IX       ;4 Byte
21BD:4      IX       ;

```

```

21BE:4 IX ;übergehen
21BF:4 IX ;
21CO:8E LPOE ;Adresse des nächsten Tokenstrings
21C1:13 4 LIQ 4 ;speichern, also
21C3:A MVB ;(OE,OF)=(04,05)
21C4:84 LP04 ;Aktuelle Adresse im Listing
21C5:13 C LIQ C ;zurückholen, also
21C7:A MVB ;(04,05)=(0C,0D)
21C8:5A LPOA ;Byte
21C9:59 LDM ;nochmal aus Zwischenspeicher holen
21CA:90 LP10 ;mit Anfang des Tokenstrings im RAM
21CB:C7 CPMA ;vergleichen
21CC:28 13 JRNZP $21E0;Sprung wenn Token mit nächstem Buchstabe erreicht
21CE:5 DX ;X=ins vor aktueller Adresse im Listing
21CF:24 IXL ;Buchstabe lesen
21DO:C7 CPMA ;mit Token im RAM vergleichen
21D1:28 8 JRNZP $21DA;Sprung wenn nicht identisch
21D3:50 INCP ;im RAM eins weiter
21D4:41 DECI ;Längenzähler erniedrigen
21D5:29 7 JRNZM $21CF;Schleifensprung bis Zähler Null
21D7:DB EXAM ;Heureka:ein Schlüsselwort gefunden, Token zum Akku
21D8:2C D JRNZP $21E6;Sprung zum Wegschreiben
21DA:84 LP04 ;X-Register mit Adresse
21DB:13 E LIQ E ;des nächsten Tokenstrings laden
21DD:A MVB ;(04,05)=(0E,0F)
21DE:2D 2F JRM $21B0;Rücksprung erneuter Vergleich
21E0:8A LPOA ;zwischen gespeichertes Byte
21E1:59 LDM ;in Akku zurückholen
21E2:84 LP04 ;X-Register wieder mit aktueller
21E3:13 C LIQ C ;Adresse im Listing laden, also
21E5:A MVB ;(04,05)=(0C,0D)
21E6:26 IYS ;wegschreiben
21E7:48 INCK ;Längenzähler erhöhen
21E8:2D 79 JRM $2170;Rücksprung nächstes Zeichen
21EA:26 IYS ;OD noch wegschreiben
21EB:48 INCK ;Längenzähler hochzählen
21EC:10 0 0 LIDP 0 ;hier steht dann die Adresse für Längenattribut
21EF:88 LPO8 ;Längenzähler
21F0:53 MVM ;in diese Adresse
21F1:2 FF LIA FF ;Akku mit Endmarke laden
21F3:26 IYS ;wegschreiben
21F4:10 46 E3 LIDP 46E3;DP=Pointer auf Basicende
21F7:86 LPO6 ;Y-Register
21F8:1B 8 EXDB ;dorthin
21F9:6B 8 TEST 8 ;Break gedrückt?
21FB:39 EE JRMZ $210E;Rücksprung nächste Zeile lesen und Tokenisieren
21FD:37 RTN ;Return zum Basic
21FE:5B POP ;verbliebenes Byte vom Stack nehmen
21FF:37 RTN ;und Return zum Basic

```

```

1 "RS232C-TREIBER FUER DEN SHARP PC-1401/02 BY BO-FU

```

```

20 DATA $84,$10,$46,$E1,$1A,$24,$67,$FF,$38,$A0,$6B,$08,$28,$9C,$34,$00
20 DATA $07,$23,$89,$1E,$5B,$DA,$C3,$3A,$0B,$02,$56,$90,$0C,$02,$02,$8F
30 DATA $0C,$2D,$0C,$24,$DA,$C3,$3A,$07,$02,$01,$90,$0C,$2D,$08,$88,$02
40 DATA $05,$DB,$C8,$23,$8E,$DB,$74,$30,$C9,$28,$06,$67,$30,$38,$05,$C8
50 DATA $78,$XX,$AB,$00,$02,$90,$1D,$49,$29,$17,$04,$02,$20,$34,$78,$XX
60 DATA $AB,$5B,$67,$0D,$28,$06,$78,$XX,$A5,$2D,$55,$24,$5A,$3A,$04,$D2
70 DATA $2D,$14,$67,$F6,$28,$03,$02,$5C,$67,$F8,$28,$03,$02,$28,$84,$74
80 DATA $C6,$DB,$34,$23,$5A,$74,$A9,$85,$DB,$34,$24,$04,$85,$DB,$84,$55
90 DATA $05,$24,$67,$7F,$2A,$06,$78,$XX,$AB,$2D,$09,$85,$5B,$DB,$84,$5B
100 DATA $DB,$2D,$47,$02,$AF,$84,$DB,$85,$02,$47,$DB,$24,$34,$78,$XX,$AB
110 DATA $5B,$67,$0D,$29,$09,$02,$0A,$2C,$03,$02,$1A,$2C,$01,$12,$5F,$61
120 DATA $40,$DF,$6B,$88,$39,$03,$00,$08,$12,$5E,$61,$03,$5F,$4E,$0C,$D2
130 DATA $2A,$14,$40,$FD,$4E,$04,$5F,$4D,$4D,$41,$29,$06,$60,$FD,$4E,$0D
140 DATA $5F,$60,$FE,$5F,$37,$61,$02,$2D,$12,$10,$46,$4E,$86,$1A,$07,$06
150 DATA $88,$55,$49,$12,$5E,$61,$01,$5F,$00,$08,$03,$00,$CC,$5A,$3B,$03
160 DATA $4E,$09,$4E,$0E,$CC,$5A,$DA,$D2,$DA,$41,$29,$09,$DA,$26,$67,$0D
170 DATA $38,$04,$49,$29,$1C,$12,$5E,$60,$FE,$5F,$37,$4D,$4D,$4D,$02,$47
180 DATA $87,$DB,$02,$60,$86,$DB,$02,$9E,$88,$DB,$78,$XX,$E2,$2A,$04,$02
190 DATA $0D,$52,$10,$46,$E3,$86,$1A,$07,$02,$47,$85,$DB,$02,$60,$84,$DB
200 DATA $00,$0F,$23,$88,$1E,$24,$75,$30,$3A,$C4,$67,$10,$24,$C0,$90,$D0
210 DATA $24,$67,$40,$2A,$1C,$67,$30,$3A,$18,$75,$30,$34,$82,$13,$10,$04,$0A
220 DATA $00,$09,$90,$14,$3A,$A9,$41,$29,$06,$03,$00,$5B,$90,$14,$2D,$1F
230 DATA $91,$DB,$26,$90,$DB,$26,$06,$10,$XY,$ED,$87,$53,$86,$11,$EE,$53
240 DATA $24,$67,$0D,$38,$76,$67,$20,$39,$08,$67,$22,$28,$0C,$4E,$48,$24
250 DATA $67,$0D,$38,$67,$67,$22,$29,$0A,$67,$41,$3A,$5B,$67,$5A,$2A,$57
260 DATA $8A,$DB,$13,$04,$8C,$0A,$84,$02,$12,$DB,$85,$02,$A9,$DB,$03,$00
270 DATA $8A,$59,$D1,$5A,$84,$14,$24,$67,$00,$38,$36,$04,$85,$DB,$84,$55
280 DATA $00,$FF,$05,$8F,$50,$04,$55,$40,$63,$80,$3B,$07,$04,$04,$04,$04
290 DATA $8E,$13,$04,$0A,$84,$13,$0C,$0A,$8A,$59,$90,$C7,$28,$13,$05,$24
300 DATA $C7,$28,$08,$50,$41,$29,$07,$DB,$2C,$0D,$84,$13,$06,$0A,$62,$2F
310 DATA $8A,$59,$84,$13,$0C,$0A,$26,$48,$2D,$79,$26,$48,$1E,$00,$88
320 DATA $53,$02,$FF,$26,$10,$46,$E3,$86,$1B,$6B,$08,$39,$EE,$47,$5B,$37
330 XX=PEEK (646E2)-2;XY=XX+1;TA=XX+256;SU=6*XX*XY+9B0D;S=0
340 FOR I=IATO IA+1:FF:READ X:5=S+S:X:POKE I,X:NEXT I
350 IF S<>SU THEN PRINT "DATA-ERROR":END
350 PAUSE "RS232C":PAUSE "INSTALLIERT"
360 PRINT "LIST# =CALL#";HEX (IA)

```

```

370 PRINT "PRINT# =CALL#";HEX (IA+69)
380 PRINT "CTRL-Z =CALL#";HEX (IA+6A9)
390 PRINT "INPUT# =CALL#";HEX (IA+6D0)
400 PRINT "MERGE# =CALL#";HEX (IA+610C)
410 INPUT "NOCHMAL?";N:GOTO 350

```

```
* Empfangsprogramm RS232C für den Sharp PC 1402 /1401  
* von Ansgar Boczek-Funcke  
* Schauenburger Str.54  
* 2300 Kiel 1
```

```
'geschrieben in Turbobasic (Borland)  
'läuft auf allen IBM-komp 8088,8086 und 80286-Rechnern  
cls  
zeit=timer  
on error goto errorhandler  
infile$=commands$  
if infile$="" then locate 2,1:input "Filename";infile$  
open infile$ for input as #2  
open "com1: 4800,n,8,2,1f,ds0,cd0,cs1000" as #1 len =1  
while not eof(2)  
  line input #2,zeile$  
  if left$(zeile$,1)=" " then zeile$=mid$(zeile$,2)  
  locate 24,1:print zeile$;  
  locate 1,1:print "in progress "  
  locate 1,40:print space$(40)  
  print #1,zeile$+chr$(6h0D)  
wend  
locate 1,1:print "finished "  
close #1  
cls  
end  
errorhandler:  
if err=24 then  
  locate 1,1:print space$(40);  
  print "Sharp not ready press ESC to resume "  
if inkey$ =chr$(27) or timer-zeit>120 then  
  cls  
  end  
end if  
resume  
else  
  PRINT "Fehler ",ERR;"in "; ERL  
end  
end if
```

```
* Empfangsprogramm RS232C für den Sharp PC 1402 /1401  
* von Ansgar Boczek-Funcke  
* Schauenburger Str.54  
* 2300 Kiel 1
```

```
'in Turbobasic (Borland)  
cls  
Scom1 10000  
on com(1) gosub byte  
on error goto fehler  
oufile$=commands$  
if oufile$="" then oufile$="sharp.bas"  
open oufile$ for output as #2  
com(1) on  
open "com1: 4800,n,8,1,ds0,cd0,cs0" as #1 len =1  
locate 1,1:Print "press escape when finished"  
do  
loop until inkey$=chr$(27)  
cls  
end  
byte:  
while not eof(1)  
  a$=input$(loc(1),#1)  
  locate 2,pos:print a$;  
  print #2,a$;  
  if instr(a$,chr$(6h1a)) then cls:end  
wend  
return  
fehler:  
if err=57 then resume else end
```

Kassette 98.- DM (incl. 14% Mwst.)

SHARP, CASIO Pocket Computer

Super-Bestellschein

Lieferschrift:

Hiermit bestelle ich:

Anzahl: Buch:

PC-1500(A)/PC-1600

- Die besten Programme für PC-1500(A)/PC-1600
- ISBN 3-624327-26-2, VK = 48,- DM
- PC-1500(A)/PC-1600 Hardwarehandbuch,
- ISBN 3-624327-13-0, VK = 48,- DM
- PC-1500(A) Tip- und Tricia-Programme
- ISBN 3-624327-12-2, VK = 49,- DM
- Ergänzungheft zum PC-1500(A) Maschinensprache-
buch, ISBN 3-624327-17-3, VK = 15,- DM
- PC-1500 Intern vom Scheitel
- VK = 56,- DM

PC-1600

- PC-1600 Systemhandbuch,
- ISBN 3-624327-31-9, VK = 48,- DM
- PC-1600 Anwendungshandbuch,
- ISBN 3-624327-55-8, VK = 48,- DM
- PC-1600 Maschinensprachehandbuch
- ISBN 3-624327-01-5, VK = 48,- DM
- PC-1600 Tip- und Tricia-Programmehandbuch,
- ISBN 3-624327-68-8, VK = 48,- DM

PC-1401(02/03/21/50/75

- PC-1401(02) Systemhandbuch,
- ISBN 3-624327-01-7, VK = 38,- DM
- PC-1401 Anwendungshandbuch,
- ISBN 3-624327-08-4, VK = 38,- DM
- PC-1401(02) Maschinensprachehandbuch,
- ISBN 3-624327-11-1, VK = 48,- DM
- PC-1403 Systemhandbuch,
- ISBN 3-624327-98-1, VK = 38,- DM
- PC-1403 Anwendungshandbuch,
- ISBN 3-624327-85-3, VK = 48,- DM
- PC-1403 Maschinensprachehandbuch,
- ISBN 3-624327-73-4, VK = 48,- DM
- PC-1450 Anwendungshandbuch,
- ISBN 3-624327-18-1, VK = 48,- DM
- PC-1450 Maschinensprachehandbuch,
- ISBN 3-624327-23-4, VK = 49,- DM
- PC-1401(02/21) Tip- und Tricia-Programmehandbuch,
- ISBN 3-624327-33-5, VK = 48,- DM
- PC-1401(02/21) Maschinenspracheprogrammierung,
- ISBN 3-624327-15-5, VK = 48,- DM
- PC-1421 Begleitheft,
- ISBN 3-624327-78-9, VK = 15,- DM
- PC-1475(PC-1280) Anwendungshandbuch,
- ISBN 3-624327-66-1, VK = 48,- DM
- PC-1260(PC-1475) Systemhandbuch
- ISBN 3-624327-89-3, VK = 48,- DM

PC-2500

- PC-2500 Systemhandbuch,
- ISBN 3-624327-20-3, VK = 49,- DM

PC-1350

An alle Auslandskunden

Wenn Sie Ihre uns bestellen, so fügen Sie bitte einen Voraussscheck (Euro-
check) bei. Sie ersetzen sich damit eine ungenutzte Gehaltskarte, da Nach-
nahmenleistungen ins Ausland sehr viel mehr kosten und auch wesentlich
länger unterwegs sind.
Das gilt auch wenn Sie z.B. in Österreich oder in der Schweiz wohnen.

Bankverbindung: Postgirokonto 461533-103
 Postgiromat Berlin-West
 Bankleitzahl: 10010010



- PC-1350 Anwendungshandbuch,
- ISBN 3-624327-15-7, VK = 48,- DM
- PC-1350 Maschinensprachehandbuch,
- ISBN 3-624327-10-4, VK = 48,- DM

PC-1245/48/51/60/61/80

- PC-1244 Programmiersammlung
- ISBN 3-624327-00-4, VK = 48,- DM
- PC-1260/81 Maschinensprachehandbuch,
- ISBN 3-624327-29-7, VK = 48,- DM
- PC-1475(PC-1260) Anwendungshandbuch,
- ISBN 3-624327-66-5, VK = 48,- DM
- PC-1260(PC-1475) Systemhandbuch
- ISBN 3-624327-66-3, VK = 48,- DM

PC-1100

- PC-1100 Anwendungshandbuch,
- ISBN 3-624327-45-9, VK = 38,- DM
- MZ 700/800
- MZ 700/800 Maschinensprachehandbuch,
- ISBN 3-624327-07-4, VK = 48,- DM

Casio FX-850P

- Casio FX-850P Anwendungshandbuch
- ISBN 3-98374-000-7, VK = 48,- DM

Sharp Taschencomputer Allgemein

- Rechnenopkop mit Sharp Taschencomputer,
- ISBN 3-624327-60-7, VK = 48,- DM
- Umsetzungshandbuch für Sharp Taschencomputer,
- ISBN 3-624327-77-1, VK = 48,- DM
- Hardware-Erweiterung für Sharp Pocket-Computer
- ISBN 3-624327-62-0, VK = 48,- DM
- Hardware-Erweiterungen für Sharp Taschencomputer,
- ISBN 3-7729-8351-3, VK = 48,- DM
- Maschinensprachehandbuch für Sharp Taschencomputer,
- ISBN 3-624327-74-2, VK = 48,- DM
- Datenübertragungshandbuch für Sharp Taschencomputer,
- ISBN 3-624327-63-7, VK = 48,- DM
- Datenfassungshandbuch für Sharp Taschencomputer,
- ISBN 3-624327-62-3, VK = 48,- DM
- CAD- und Grafikprogrammiersammlung für Sharp
Taschencomputer, ISBN 3-624327-44-0, VK = 48,- DM
- Basic-Erweiterungen für Sharp Taschencomputer,
- ISBN 3-624327-40-8, VK = 48,- DM
- Schreib- und Testverfahren für Sharp Computer,
- ISBN 3-624327-37-8, VK = 48,- DM
- Hi-Calc-Handbuch für Sharp Computer,
- ISBN 3-624327-24-6, VK = 48,- DM
- Computerlexikon und Rechnerführer für Sharp
Computer, ISBN 3-624327-21-1, VK = 48,- DM
- Lehr- und Übungshandbuch für Sharp Taschencomputer
- ISBN 3-624327-09-2, VK = 48,- DM
- Basic-Lehrbuch für Sharp Computer,
- ISBN 3-624327-08-1, VK = 48,- DM
- Grafikhandbuch für Sharp Computer,
- ISBN 3-624327-04-1, VK = 48,- DM
- Flugnavigation mit Sharp Taschencomputer
- ISBN 3-624327-78-2, VK = 48,- DM
- Navigationsprogrammiersammlung für Sharp Computer,
- ISBN 3-624327-49-1, VK = 48,- DM
- Kaufmännische Programmiersammlung für Sharp Taschen-
computer, ISBN 3-624327-75-0, VK = 48,- DM
- Betriebswirtschaft mit Sharp Taschencomputern,
- ISBN 3-624327-69-8, VK = 48,- DM
- Finanz- und Wirtschaftsprüfungssammlung für Sharp
Computer, ISBN 3-624327-30-0, VK = 48,- DM
- Finanz- und Wirtschaftsprüfungssammlung für Sharp

- Computer, Band 2, ISBN 3-624327-07-1, VK = 48,- DM
- Wertpapierverwaltung mit Sharp Taschencomputer,
- ISBN 3-624327-60-2, VK = 48,- DM
- Steuerprogrammiersammlung für Sharp Taschen-
computer, ISBN 3-624327-51-3, VK = 48,- DM
- Lohn- und Erlösrechner mit Sharp Taschencomputer
- ISBN 3-624327-46-3, VK = 48,- DM
- Fremdsprachehandbuch für Sharp Taschencomputer,
- ISBN 3-624327-76-9, VK = 48,- DM
- Mathematikprogrammiersammlung für Sharp Computer,
- Band 1, ISBN 3-624327-25-4, VK = 48,- DM
- Mathematikprogrammiersammlung für Sharp Computer,
- Band 2, ISBN 3-624327-34-3, VK = 48,- DM
- Mathematikprogrammiersammlung für Sharp Computer
- Band 3, ISBN 3-624327-60-4, VK = 48,- DM
- Statistikprogrammiersammlung für Sharp Computer,
- ISBN 3-624327-34-3, VK = 48,- DM
- Lehr- und Schulprogrammiersammlung für Sharp Computer
- ISBN 3-624327-56-0, VK = 38,- DM
- Elektronische Programmiersammlung für Sharp Taschen-
computer, ISBN 3-624327-46-7, VK = 48,- DM
- Elektronische für Sharp Taschencomputer, Band 2
- ISBN 3-624327-003-1, VK = 48,- DM
- Metallartenverarbeitung mit Sharp Taschencomputer,
- ISBN 3-624327-72-6, VK = 48,- DM
- Chemieprogrammiersammlung für Sharp Taschencomputer,
- Band 1, ISBN 3-624327-70-3, VK = 48,- DM
- Chemieprogrammiersammlung für Sharp Taschencomputer,
- Band 2, ISBN 3-624327-81-0, VK = 48,- DM
- Programmiersammlung für Maschinen mit Sharp Taschen-
computern, ISBN 3-98374-022-3, VK = 48,- DM
- Physikprogrammiersammlung für Sharp Taschencomputer,
- ISBN 3-624327-43-2, VK = 48,- DM
- Vermessungsprogrammiersammlung für Sharp
Taschencomputer, Band 1
- ISBN 3-624327-55-2, VK = 48,- DM
- Einbaugewinn und Bauteile-Programmiersammlung für Sharp
Computer, ISBN 3-624327-41-0, VK = 48,- DM
- Programmiersammlung für Heizungs-, Klima- und Sanitär-
technik mit Sharp Taschencomputern
- ISBN 3-624327-66-6, VK = 48,- DM
- Fototechnik mit Sharp Taschencomputern,
- ISBN 3-624327-66-5, VK = 48,- DM
- Tom- und Musikprogrammiersammlung für Sharp Taschen-
computer, ISBN 3-624327-83-1, VK = 48,- DM
- Schachprogrammiersammlung für Sharp Taschencomputer,
- ISBN 3-624327-64-5, VK = 48,- DM
- Lotto- und Glücksspiel-Programmiersammlung für Sharp-
Taschencomputer, ISBN 3-624327-62-6, VK = 48,- DM
- 101 Spiele für Sharp Taschencomputer,
- ISBN 3-624327-54-8, VK = 38,- DM
- Spiele für Sharp Taschencomputer, Band 2
- ISBN 3-624327-67-4, VK = 48,- DM
- Software-Flucht,
- ISBN 3-624327-03-3, VK = 38,- DM

Gesamtpreis: DM

Datum, Unter-
schrift:

Probehett der Zeitschrift "POCKET COMPUTER" gegen
 6,- DM in Briefmarken
 (aus dem Ausland 7,- DM Eurocheck oder internationale Postwertzeichen)

Name : _____
 Vorname : _____
 Straße : _____
 Plz. : _____ Ort : _____

Datum : _____ Unterschrift : _____

einsenden an
 Pocket Computer
 Fischel GmbH
 Kaiser Friedrich Str. 54a
 D-1000 Berlin 12
 Tel. : 030 / 323 60 29

DURCH INFORMATION VORN

Do not sale!



Sehr geehrter Software Lieferant !

Für eine Zusammenarbeit gibt es drei Möglichkeiten :

- a) Listings, Fachbeiträge und Programme auf Diskette oder Kassette.
Sendungen werden nur bearbeitet, wenn eine unterschiedene Überlassungserklärung beiliegt.
- b) Produktbeschreibungen für Soft- und Hardware.
- c) Anzeigen

zu a) Überlassung

- 1) Der Einreicher versichert, daß seine Sendung frei von Urheberrechten Dritter ist.
- 2) Programme sind als Listing und auf Datenträger (Diskette oder Kassette) einzureichen. Eine druckreife Beschreibung muß beiliegen.
- 3) Grundsätzlich übernimmt der Verlag keine Haftung und Verpflichtung für Sendungen jeglicher Art.
- 4) Die Einsendung gilt als Zustimmung zum Abdruck. Einer gesonderten Zustimmung des Verfassers bedarf es nicht.
- 5) Mit der Annahme der Sendung erwirbt der Verlag vom Überlasser sämtliche Rechte, einschließlich der Vervielfältigung zu gewerblichen Zwecken.
- 6) Der Überlasser erhält eine Gratifikation nach Ermessen des Verlages. Der Überlasser kann Gratifikationswünsche angeben, bzw. vorher aushandeln.
Der Überlasser erkennt mit seiner Unterschrift diese Bedingungen der FISCHEL GmbH, Kaiser Friedrich Str. 54a, 1000 Berlin 12, an.

.....
Datum

.....
Unterschrift des Einreichers

zu b) Provision

- 1) Bestellungen für Software von dürfen von der Fischel GmbH, Kaiser Friedrich Str. 54a, 1000 Berlin 12 entgegen genommen werden.
- 2) Bestellungen müssen an:(Name,Strasse,Plz,Ort) weitergeleitet werden.
- 3) Für die Entgegennahme von Bestellungen und deren Weiterleitung erhält die Fischel GmbH je Bestellung ein Brutto Entgelt (die gesetzliche Mehrwertsteuer ist bereits enthalten) in Höhe von 25 % (fünfundzwanzig Prozent) des Bruttoverkaufspreises (VK = inkl. Mwst.). Die Abrechnung der Provision erfolgt durch Scheck nach Zahlungseingang beim Anbieter oder bei Vorkasse durch Abzug.
- 4) Diese Vereinbarung kann zum Ende eines jeden Kalenderquartals mit einer Frist von einem Monat gekündigt werden.
- 5) Im Falle eines Rechtsstreites ist Berlin Gerichtsstand.

verbindliche Unterschrift

.....
Ort, Datum

.....
Unterschrift des Anbieters

zu c) Anzeigen in "Pocket Computer"

- 1) Kleinanzeigen

Wenn Sie eine Anzeige aufgeben wollen, senden sie uns bitte den unten folgenden Abschnitt mit einem Scheck über den zu leistenden Betrag zu, sonst wird Ihre Kleinanzeige nicht bearbeitet. Wir behalten uns das Recht vor, Kleinanzeigen ohne Angabe von Gründen abzulehnen. Eingereichte Schecks werden dann natürlich umgehend zurückerstattet.
Bitte ausfüllen, ausschneiden und an die Fischel GmbH senden.

Bitte veröffentlichen Sie in der nächst erreichbaren Ausgabe von "Pocket Computer" den folgenden Text:

Bitte nur 30 Zeichen pro Zeile einschließlich Satzzeichen und Wortzwischenräumen eintragen. Jede angefangene Zeile kostet 6,50 DM (incl 14% Mwst). Der Gesamtbetrag liegt als Scheck bei.

Name:

Straße, Nr:

PLZ, Ort:

Telefon(für Rückfragen):

Datum, Unterschrift:

- 2) Produktanzeigen

Es gilt die Anzeigenpreisliste von Juli 1988. Weitere Informationen auf Anfrage. Wir werden Ihnen gerne ein kostengünstiges Angebot machen.

Do not sale !



ISBN: 3-924327-63-7

Sharp Microcomputer
 Fischel GmbH
 Kaiser-Friedrich-Str. 54 a
 D - 1000 Berlin 12
 Tel. 030 / 323 60 29
 Mo - Fr 10 - 18.30, Sa - 14 h

Do not sale !